

BABD: A Bitcoin Address Behavior Dataset for Address Behavior Pattern Analysis

Yuxin Xiang^{1,*}, Yuchen Lei^{1,*}, Ding Bao^{1,*}, Wei Ren^{1,2,3,†}, Tiantian Li¹, Qingqing Yang¹, Wenmao Liu⁴, Tianqing Zhu⁵, and Kim-Kwang Raymond Choo⁶

¹ School of Computer Science, China University of Geosciences, Wuhan 430074, China

² Henan Key Laboratory of Network Cryptography Technology, Zhengzhou 450001, China

³ Hubei Key Laboratory of Intelligent Geo-Information Processing, Wuhan 430074, China

⁴ NSFOCUS Technologies Group Co.,Ltd.

⁵ School of computer science, University of Technology Sydney, Sydney, Australia

⁶ Department of Information Systems and Cyber Security, University of Texas at San Antonio
San Antonio, TX 78249-0631, USA

{yuxin.xiang,root,ding.bao,weirencs,ltt,qingqing.yang}@cug.edu.cn

liwenmao@nsfocus.com,tianqing.zhu@uts.edu.au,raymond.choo@fulbrightmail.org

ABSTRACT

Cryptocurrencies are no longer just the preferred option for cybercriminal activities on darknets, due to the increasing adoption in mainstream applications. This is partly due to the transparency associated with the underpinning ledgers, where any individual can access the record of a transaction record on the public ledger. In this paper, we build a dataset comprising Bitcoin transactions between 12 July 2019 and 26 May 2021. This dataset (hereafter referred to as BABD-13) contains 13 types of Bitcoin addresses, 5 categories of indicators with 148 features, and 544,462 labeled data. We then use our proposed dataset on common machine learning models, namely: k -nearest neighbors algorithm, decision tree, random forest, multilayer perceptron, and XGBoost. The results show that the accuracy rates of these machine learning models on our proposed dataset are between 93.24% and 96.71%. We also analyze the proposed features and their relationships from the experiments, and propose a k -hop subgraph generation algorithm to extract a k -hop subgraph from the entire Bitcoin transaction graph constructed by the directed heterogeneous multigraph starting from a specific Bitcoin address node (e.g., a known transaction associated with a criminal investigation).

1 INTRODUCTION

Cryptocurrencies, such as Bitcoin, remain increasingly popular. For example, according to CoinMarketCap, the global cryptocurrency market capital is \$1.77 Trillion U.S. dollars and Bitcoin's market capital is estimated to be at \$0.74 Trillion U.S. dollars (as of February 27, 2022 - see <https://coinmarketcap.com>). Similar to fiat currencies, there are concerns about the misuse of cryptocurrencies (e.g., cyber-criminal and darknet markets). However, unlike fiat currencies, one can more easily trace cryptocurrency transactions partly because of the transparency nature of the public ledger.

It is, therefore, unsurprising that there have been numerous attempts to design techniques to facilitate cryptocurrency transaction tracing [1, 16, 25, 28, 29, 32, 33]. For example, in the context of Bitcoin transactions, one could utilize graph analysis to determine or classify different address types. There are, however, a number of challenges and limitations in existing approaches, ranging from accuracy to information loss to (in)completeness (e.g., not considering internal relations and differences of different classifications). Another limitation we observe is the lack of a comprehensive dataset that can be used as a baseline for the research community.

Therefore, in this paper, we present a general framework that can be used to build a Bitcoin transaction graph by the directed heterogeneous multigraph and analyze address nodes from the graph. We then collect and compile Bitcoin transactions that occur between 12 July 2019 and 26 May 2021 into a dataset (hereafter referred to as BABD-13). The latter has 13 types of Bitcoin addresses, 544,462 labeled data, and 5 categories of indicators with 148 features. We then use BABD-13 as the baseline dataset to evaluate common machine learning models (i.e., k -nearest neighbors algorithm, decision tree, random forest, multilayer perceptron, and XGBoost). Findings from the evaluations suggest that the accuracy obtained ranges between 93.24% and 96.71%. To extract the structural features, we also propose a novel method to generate k -hop subgraph for an address node from the entire Bitcoin transaction graph built by the directed heterogeneous multigraph.

In the next section, we will briefly review the extant literature.

2 RELATED WORK

As discussed earlier, a number of approaches to analyze Bitcoin transactions have been proposed in the literature, such as those focusing on quantitative studies [28]. For example, using Bitcoin transaction data from 2009 to 2014, Alqassem et al. [1] studied the evolution of graph structural properties over time. They observed that the Bitcoin transaction graph is generally similar to typical social networks in the structural index. To understand the structural features of the Bitcoin transaction graph, Popuri and Gunes [25] measured the general characteristics of the Bitcoin network using the complex network theory. Similarly, Serena et al. [29] considered

* The first three authors contributed equally to this work.

† Corresponding author at: School of Computer Science, China University of Geosciences, Wuhan, China. Email: weirencs@cug.edu.cn.

the cryptocurrency transaction graph as a complex network and studied the Bitcoin transaction graph using complex network-based methods. Specifically, using degree distribution and aggregation clustering coefficient, they calculated several simple indexes and their changes [29]. More recently in 2021, Tao et al. [33] proposed and implemented a complex network-based framework to comprehensively analyze the Bitcoin transaction graph. They observed that the non-rich club effect, small-world phenomenon, and other typical characteristics exist in the Bitcoin transaction graph.

Due to the increasing popularity of cryptocurrencies, there has been a corresponding increase in such currencies from governments and regulatory agencies. This reinforces the importance of understanding the full Bitcoin transaction graph so that governments and regulatory agencies (e.g., taxation, law enforcement, and financial intelligence units, as well as anti-money laundering/counter-terrorism financing regulators) can monitor and trace criminal proceeds. In other words, how do we more accurately distinguish normal from suspicious cryptocurrency addresses? Such addresses generally include exchange-related addresses and mining pool-related addresses.

Ranshous et al. [26] used the directed hypergraph to analyze the patterns associated with exchange addresses, and proposed different types of short thick bands (STBs) to identify the patterns. In addition, they also applied basic machine learning methods to classify early exchange addresses. Focusing on mining pools (another kind of normal pattern), Romiti et al. [27] selected three of four the biggest pools to empirically investigate the relationships among them. They mainly utilized economic activity pattern-based ways to explore the relationships of miner-owned addresses and pools. In a separate work, Tovanich et al. [34] studied pool hopping behaviors in 15 pools of Bitcoin transactions. Based on the empirical study and their proposed heuristic algorithm designed to describe the payout flows, they determined that pool fees and payout schemes are the two most important factors to influence the behaviors of miner-owned addresses.

There have also been attempts to detect activities associated with illicit addresses or transactions [11, 22, 37, 38]. For example, Liao et al. [12] measured data such as Bitcoin amount in ransomware cases and the resulting financial loss. Conti et al. [7] also proposed an approach to analyzing the financial impact of ransomware and presented the timeline of the ransomware process. They then built a public dataset of ransomware-related Bitcoin addresses. Additionally, Paquet-Clouston et al. [21] proposed an efficient framework to identify and collect ransomware attack-related Bitcoin transaction addresses. Their research demonstrated the change of the Bitcoin amount over time from 2013 to 2017. Bartoletti et al. [2] collected Ponzi scheme Bitcoin addresses from various Bitcoin forums, and based on their analyses they proposed 11 features for Bitcoin address classification. Vasek and Moore [36] obtained their Ponzi scheme data from a bitcoin forum¹ and observed that cybercriminals interacted more frequently with the victims and posted more comments in the thread. In addition, Toyoda et al. [35] presented a systematic approach to study the high yield investment program (HYIP), which is a kind of Ponzi scheme. They analyzed the HYIP addresses based on historic records and applied supervised learning

to classify the HYIP addresses. The accuracy rate of their approach is as high as 93.75%.

There have also been approaches designed to learn the similarities and differences among different kinds of Bitcoin addresses or transactions in the Bitcoin transaction graph [5, 11, 13, 15, 16, 42]. However, most existing approaches prefer to utilize the simplified Bitcoin transaction graph (e.g., undirected simple graph), rather than its original structure, in their analysis [22]. Consequently, this may result in information loss.

The types of different addresses in most existing works are inaccurate and/or incomplete. For instance, at most five types of addresses were analyzed together in recent works such as that of [11]. This is clearly not sufficient in having an in-depth understanding of the behavior patterns in the Bitcoin transaction graph.

In addition, the indicators utilized for analysis proposed in existing approaches, such as those of [5, 15, 16, 42], are generally not systematic and/or comprehensive. Specifically, they do not classify their indicators and do not include key indicators that can be extracted from the Bitcoin transaction graph. Besides, these approaches are hard to replicate since it is not clear (publicly) how the Bitcoin transaction graph(s) is/are built and generated.

In the next section, we will briefly introduce the relevant background materials.

3 BACKGROUND MATERIALS

3.1 Bitcoin Graph Structure

Bitcoin transaction graph is essentially a kind of unspent transaction output (UTXO) model. Aiming at different kinds of issues existing in the huge Bitcoin network, various Bitcoin network models are proposed by researchers [13, 39]. For general analysis on the Bitcoin network, transaction nodes or address nodes are main objects [2, 11, 15, 41], especially transaction nodes because they include more information. Besides, there are also a great number of works that regard related addresses as an entity for analysis [5, 40].

However, in our view, the above and similar approaches to simplify the Bitcoin transaction graph will lead to information loss to the Bitcoin graph analysis. In this paper, for diminishing the information loss of the Bitcoin network as much as possible while analyzing the transaction patterns, we proposed an improved directed heterogeneous multigraph Bitcoin structure based on the structure proposed by Maesa et al. [8], which includes characteristics of both the address (*Ads*) node and the transaction (*Tx*) node to ensure the accuracy of the analysis. Our designed structure is shown in Fig. 1 in detail.

3.2 Behavior Classifications and Definitions

It is significant to classify and define typical behaviors of Bitcoin addresses exactly. Specifically, the concepts and basic motivations of different illicit transactions will help us improve the effect of classification tasks for Bitcoin addresses and understand deeper the patterns that exist in Bitcoin address behaviors.

We will study and analyze the mainstream types of illicit transactions in this paper, which are listed below:

- (1) **Blackmail.** Cryptocurrency blackmail has three typical categories that are ransomware, sextortion, and scam. Apart

¹<https://bitcointalk.org>

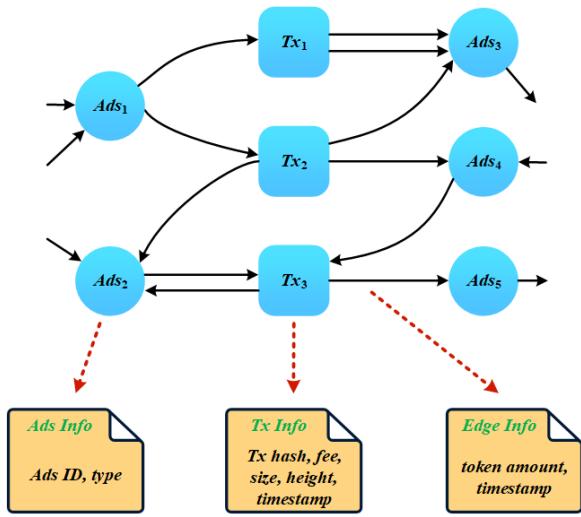


Figure 1: Bitcoin transaction graph structure

from the blackmail methods, these blackmail types are similar in most aspects, which utilize some ways to threat or deceive the victims to pay a certain amount of cryptocurrency to several specific addresses.

- (2) **Cyber-Security Service.** Cyber-security services denote that the providers can offer payment gateways, proxy or virtual private network (VPN) services, and other cyber-security-related services. In this case, providers only accept cryptocurrency as payment in order to improve the security of their services.
- (3) **Darknet Market.** Darknet markets are the markets hiding in the darknet, where people are able to buy and sell illegal stuff and services such as automatic rifles and assassination services anonymously. Besides, in order to enhance the anonymity of these dirty transactions, traders are oriented to exchange through cryptocurrencies.
- (4) **Centralized Exchange.** Centralized cryptocurrency exchanges earn fees by acting as trustworthy intermediaries among their customers. They execute the “Know Your Customer (KYC)” policy and allow their customers to trade cryptocurrency/cryptocurrency pairs (e.g., BTC/USDT and ETH/BTC pairs) and do swaps between cryptocurrency and fiat.
- (5) **P2P Financial Infrastructure Service.** P2P financial infrastructure services are the P2P financial activities that conduct only by cryptocurrency. Examples are bond markets and P2P lending platforms.
- (6) **P2P Financial Service.** P2P financial applications use cryptocurrency as a reward according to users’ contributions to encourage users to complete more missions that are helpful to the organizers such as increasing the number of clicks of a concrete advertisement. This kind of service includes faucets, video sharing, and affiliate marketers.
- (7) **Gambling.** Cryptocurrency gambling denotes playing casino games, such as blackjack and roulette, which only cryptocurrencies can be utilized as wagers.

- (8) **Government Criminal Blacklist.** Government criminal blacklist includes cryptocurrency addresses highly suspected or confirmed involved in criminal activities according to different countries’ laws.
- (9) **Money Laundering.** Money Laundering is to trade the dirty cryptocurrency in dirty addresses obtained from illegal transactions or activities to normal addresses in several complicated ways that others are hard to trace. Generally, money laundering includes the following three stages that are placement, laundering, and integration.
- (10) **Ponzi Scheme.** Ponzi schemes obey the rules that the initiator of the Ponzi scheme will pay high interests by cryptocurrency to the former investors through the investments from current involving investors. Therefore, the investors will believe that the initiator can easily earn profits to them in a related short time.
- (11) **Mining Pool.** Cryptocurrency mining pools consist of a large number of miners who contribute their computational resources together for improving the possibility of finding a new block to gain the reward of cryptocurrencies. Thus, the majority of addresses related to coinbase transactions in Bitcoin belong to mining pools.
- (12) **Tumbler.** Cryptocurrency tumbler is a kind of service that prevents monitors or censors from tracing cryptocurrency flows according to the transparent ledger, which increases the anonymity of cryptocurrencies. Generally, the service providers combine multiple inputs during a long period of time and send different inputs to their planned destinations at random times to implement cryptocurrency tumblers.
- (13) **Individual Wallet.** Individual wallets stand for the wallets owned by ordinary people who use cryptocurrency as a payment way in daily life, such as shopping and dining.

3.3 Data Collection

There are two kinds of data we need to collect respectively that are Bitcoin ledger data and Bitcoin address data with labels. We will illustrate how we gather them in this section.

Bitcoin Ledger. We collect the Bitcoin ledger using the public API². Our current research Bitcoin ledger data is based on 100,001 blocks data, where the block height is from 585,000 to 685,000 (from July 12, 2019, to May 26, 2021), including 516,167,131 address and transaction nodes and 713,703,239 relationship edges.

Bitcoin Address. Compared to the Bitcoin ledger, it is harder to gather Bitcoin addresses with labels. In order to collect more data while keeping the accuracy of the data, we divide our collected data into two types are “strong address (SA)” and “weak address (WA)”, where SA denotes the address with the strongly confirmed label, in contrast, WA means the address with the week confirmed label (e.g. the reported address). SAs mainly collected from the public dataset³ and government blacklists⁴. Furthermore, we also find valuable SA data from the high influential papers [2, 7]. Besides, WAs are gathered from the public reported Bitcoin address dataset⁵.

²<https://btc.com>

³<https://www.walletexplorer.com>

⁴<https://home.treasury.gov>

⁵<https://www.bitcoinabuse.com>

and a number of social accounts on Twitter⁶. All the data collected is listed in Table 1.

Table 1: Labeled data collection

Type	Number (SA / WA)
Blackmail	8,686 (8 / 8,678)
Cyber-Security Service	91,617 (91,617 / 0)
Darknet Market	13,861 (13,861 / 0)
Centralized Exchange	300,000 (300,000 / 0)
P2P Financial Infrastructure Service	180 (180 / 0)
P2P Financial Service	9,309 (9,309 / 0)
Gambling	105,257 (105,257 / 0)
Government Criminal Blacklist	27 (27 / 0)
Money Laundering	16 (16 / 0)
Ponzi Scheme	15 (8 / 7)
Mining Pool	1,580 (1,580 / 0)
Tumbler	12,412 (10,817 / 1,595)
Individual Wallet	1,502 (52 / 1,450)
Total Number	544,462 (532,732 / 11,730)

4 PROPOSED SCHEME

4.1 Framework

Referring to the indicators and classification methods presented in several high influential related works [5, 9, 13–16, 42], we proposed our own framework for extracting and analyzing Bitcoin address behaviors from Bitcoin transaction graph that constructed using the directed heterogeneous multigraph. The proposed framework consists of two parts that are the *statistical indicator (SI)* and the *local structural indicator (LSI)* respectively.

4.2 Statistical Indicator

SI in our proposed scheme is divided into four concrete indicator types, which we will briefly introduce as follows:

- **Pure Amount Indicator (PAI).** *PAI* is related to the token amount attributes of *Ads* nodes in the Bitcoin transaction graph.
- **Pure Degree Indicator (PDI).** *PDI* includes the degree-related attributes of *Ads* nodes in the Bitcoin transaction graph.
- **Pure Time Indicator (PTI).** *PTI* is time-related attributes of *Ads* nodes.
- **Combination Indicator (CI).** *CI* is the combinations of the features from *PAI*, *PDI*, and *PTI*.

The basic *SI* is shown in Table 2 that includes *PAI*, *PDI*, and *PTI*, and features from *CI* are illustrated in Table 3. In these tables, there are three points that we need to further explain. First, the token amount in the above tables is the original data of the Bitcoin transactions, i.e., BTC. Second, the basic unit mentioned in the above tables is the solar day of 24 hours. Finally, for the features of *Ads* node in the above tables, we also compute their simplified features.

⁶<https://twitter.com>

That means we re-compute the same features after combining the same directed edges in the Bitcoin transaction graph with their attributes and make them newly added features⁷ (i.e., using PAIa11-R1 instead of PAIa11-1).

4.3 Local Structural Indicator

Transactions connect addresses in Bitcoin to form a graph, so we believe that considering the characteristics of a subgraph composed of addresses that are close to an address can reflect some information about that address. Unlike the usual $k - hop$ subgraphs, we want to include nodes that are close to each other in the corresponding undirected graph, while reflecting the real network structure. Therefore, we propose the following Algorithm 1 to generate k -hop subgraph G_k for each labeled *Ads* node and then obtaining useful *LSI*.

First, we treat the original graph as an undirected graph and traverse each edge in breadth-first order starting from a given address, after which the nodes connected to that edge are renumbered and added to the resulting subgraph in the correct direction by checking the actual direction of that edge in the original graph. In addition, for preventing G_k too large, we set two thresholds for the proposed algorithm. One threshold is the maximum value of k and another threshold is the maximum number of *Ads* nodes and *Tx* nodes.

It is noted that in our implemented Algorithm 1, for the first threshold, we choose 3,000 as the maximum number of the sum of *Ads* node and *Tx* node, and for another one, we select $k = 4$ as the maximum value. We will explain the reason below using Fig. 2.

In Fig. 2⁸, there are 8 figures of G_k with different numbers of nodes generated from the same *Ads*, from which it is clear that Fig. 2(a), Fig. 2(b), and Fig. 2(c) contain less structural information than other 5 figures of G_k . Besides, Fig. 2(g) and Fig. 2(h) have too much information that is similar to the entire transaction graph that can not easily get valuable distinctive information. Compared among the rest Fig. 2(d), Fig. 2(e), and Fig. 2(f) we observe that their structural are almost identical, thus, we choose Fig. 2(d) (i.e., G_k with 3,000 nodes) considering the efficiency of generate G_k . It is noted that we find similar conclusions in many other address nodes in different types of Bitcoin addresses during our plenty of experiments. Also, through the experiments, we find this set of hyperparameters achieves a balance between our device’s performance limit and the effect of indicators from the local Bitcoin transaction graph. The specific features are shown below:

- **Average Degree (S1).** $G_k^{\bar{d}}$ means the average degree [3] of the nodes in G_k :

$$G_k^{\bar{d}} = \frac{1}{G_k^A + G_k^T} \left(\sum_{i=1}^{G_k^A} A_i + \sum_{j=1}^{G_k^T} T_j \right), \quad (1)$$

where G_k^A is the number of *Ads* nodes, G_k^T is the number of *Tx* nodes, A_i is the degree of the *Ads* node, and T_j is the degree of the *Tx* node. Specifically, we calculate the average values and standard deviations of in-degree/out-degree/in-degree and out-degree in G_k as features in *LSI*.

⁷<https://www.kaggle.com/datasets/lemonx/babd13>

⁸Using *Ads* node 1BtBoSSnqe8mFJCUEyCNmo3EcF8Yzhpnc from Ponzi

Table 2: Basic statistical indicator

Category	Notation	Description	Dataset
PAI	$A_a^{in/out}$	The input/output token amount of an address node	PAIa1
	$A_{a,all}^{in/out}$	The total input/output token amount of an address node	PAIa11
	DAa_1	The difference of $A_{a,all}^{in} - A_{a,all}^{out}$	PAIa12
	RAa_1	The ratio of $\frac{A_{a,all}^{in}}{A_{a,all}^{out}}$	PAIa13
	$A_{a,min/max}^{in/out}$	The minimum/maximum input/output token amount of an address node	PAIa14
	$DAa_2^{in/out}$	The difference of $A_{a,max}^{in} - A_{a,min}^{in}$ and $A_{a,max}^{out} - A_{a,min}^{out}$	PAIa15
	$RAa_2^{in/out}$	The ratio of $\frac{DAa_2^{in}}{A_{a,all}^{in}}$ and $\frac{DAa_2^{out}}{A_{a,all}^{out}}$	PAIa16
	$A_{a,std}^{in/out/all}$	The standard deviation of all input/output/input and output token amounts of an address node	PAIa17
	$RAa_3^{in/out}$	The ratio of $\frac{A_a^{in}}{A_{a,all}^{in}}$ and $\frac{A_a^{out}}{A_{a,all}^{out}}$	PAIa2
	$RAa_{3,min/max}^{in/out}$	The minimum/maximum input/output value of $RAa_3^{in/out}$	PAIa21
PDI	$RAa_{3,std}^{in/out}$	The standard deviation of all input/output values of $RAa_3^{in/out}$	PAIa22
	$A_d^{in/out/all}$	The in-degree/out-degree/in-degree and out-degree of an address	PDIa1
	$RAD_1^{in/out}$	The ratio of $\frac{A_d^{in}}{A_d^{all}}$ and $\frac{A_d^{out}}{A_d^{all}}$	PDIa11
	RAd_2	The ratio of $\frac{A_d^{in}}{A_d^{out}}$	PDIa12
	DAd_1	The difference of $A_d^{in} - A_d^{out}$	PDIa13
PTI	A_p^l	The life period of an address	PTIa1
	A_p^v	The active period of an address	PTIa2
	RAp_1	The ratio of $\frac{A_p^v}{A_p^l}$	PTIa21
	A_n^v	The number of active times of an address in each basic unit of the active period	PTIa3
	$A_{n,min/max/avg}^v$	The minimum/maximum/average active times of A_n^v	PTIa31
	DA_1^v	The difference of $A_{n,max}^v - A_{n,min}^v$	PTIa32
	$A_{n,std}^v$	The standard deviation of A_n^v	PTIa33
	A^{ti}	The time interval of transactions of an address (chronological order)	PTIa4
	$A_{min/max/avg}^{ti}$	The minimum/maximum/average time interval of A^{ti}	PTIa41
	DA_1^{ti}	The difference of $A_{max}^{ti} - A_{min}^{ti}$	PTIa42
	A_{std}^{ti}	The standard deviation of time interval of A^{ti}	PTIa43

- **Degree Distribution (S2).** $G_k^{D(d)}$ is the degree distribution [3] that means the proportion of the node with the d degree in G_k [3]. Specifically, the degree here can be divided into three types that are in-degree, out-degree, and in-degree and out-degree. Besides, we select the maximum value of degree distribution as the feature of *LSI*.
- **Degree Correlation (S3).** $G_k^{C_d}$ is the degree correlation [3] of G_k that describes the relationship between nodes with large degree and nodes with small degree. In this paper, simplified Pearson degree correlation proposed by Newman

[17] is selected to measure G_k :

$$G_k^{C_d} = \frac{\frac{1}{E} \sum_{e_{ij}} d_i d_j - [\frac{1}{E} \sum_{e_{ij}} \frac{1}{2}(d_i + d_j)]^2}{\frac{1}{E} \sum_{e_{ij}} (d_i^2 + d_j^2) - [\frac{1}{E} \sum_{e_{ij}} \frac{1}{2}(d_i + d_j)]^2}, \quad (2)$$

where the edge e_{ij} connects $node_i$ and $node_j$ then d_i and d_j stand for these two nodes' degree respectively, E is the total number of edges.

Table 3: Combination indicator

Category	Notation	Description	Dataset
PAI + PDI	$RA_1^{in/out}$	The ratio of $\frac{A_{all}^{in}}{A_p^{in}}$ and $\frac{A_{all}^{out}}{A_p^{out}}$	CI1a1
	RA_2	The ratio of $\frac{DA_1}{DAd_1}$	CI1a2
PAI + PTI	$Av_a^{in/out}$	The input/output token amount of an address in each basic unit of the active period	CI2a1
	$Av_{aavg}^{in/out}$	The average input/output token amount of an address in each basic unit of the active period	CI2a11
	$Av_{amin/max}^{in/out}$	The minimum/maximum input/output token amount of an address in each basic unit of the active period	CI2a12
	$RAp_2^{in/out}$	The ratio of $\frac{Av_a^{in}}{A_p^l}$ and $\frac{Av_a^{out}}{A_p^l}$	CI2a2
	$RAp_2^{in/out}$	The average value of $RAp_2^{in/out}$	CI2a21
	$RAp_2^{min/max}$	The minimum/maximum value of $RAp_2^{in/out}$	CI2a22
	RAp_2^{std}	The standard deviation of $RAp_2^{in/out}$	CI2a23
	$\Delta A_{all}^{in/out}$	The change of $A_{all}^{in/out}$	
	$\Delta RA_1^{in/out}$	The ratio of $\frac{\Delta A_{all}^{in}}{A_t^{in}}$ and $\frac{\Delta A_{all}^{out}}{A_t^{out}}$	CI2a3
	$\Delta RA_{1avg}^{in/out}$	The average value of $\Delta RA_1^{in/out}$	CI2a31
	$\Delta RA_{1min/max}^{in/out}$	The minimum/maximum value of $\Delta RA_1^{in/out}$	CI2a32
	$\Delta RA_{1std}^{in/out}$	The standard deviation of $\Delta RA_1^{in/out}$	CI2a33
PDI + PTI	$Av_d^{in/out}$	The in-degree/out-degree of an address in each basic unit of the active period	CI3a1
	$Av_{davg}^{in/out}$	The average in-degree/out-degree of an address in each basic unit of the active period	CI3a11
	$Av_{dmin/max}^{in/out}$	The minimum/maximum in-degree/out-degree of an address in each basic unit of the active period	CI3a12
	$RAv_1^{in/out/all}$	The ratio of $\frac{Av_d^{in}}{A_p^l}$, $\frac{Av_d^{out}}{A_p^l}$, and $\frac{Av_d^{all}}{A_p^l}$	CI3a2
	$RAv_1^{in/out/all}$	The average value of $RAv_1^{in/out/all}$	CI3a21
	$RAv_1^{min/max}$	The minimum/maximum value of $RAv_1^{in/out/all}$	CI3a22
	RAv_1^{std}	The standard deviation of $RAv_1^{in/out/all}$	CI3a23
	$\Delta RA_2^{in/out}$	The ratio of $\frac{\Delta A_{all}^{in}}{A_t^{in}}$ and $\frac{\Delta A_{all}^{out}}{A_t^{out}}$	CI3a3
	$\Delta RA_{2avg}^{in/out}$	The average value of $\Delta RA_2^{in/out}$	CI3a31
	$\Delta RA_{2min/max}^{in/out}$	The minimum/maximum value of $\Delta RA_2^{in/out}$	CI3a32
	$\Delta RA_{2std}^{in/out}$	The standard deviation of $\Delta RA_2^{in/out}$	CI3a33
PAI + PDI	RA_3^{in}	The ratio of $\frac{RA_1^{in}}{A_p^l}$	CI4a1
	RA_{3avg}^{in}	The average value of RA_3^{in}	CI4a11
	$RA_{3min/max}^{in}$	The minimum/maximum value of RA_3^{in}	CI4a12
	RA_{3std}^{in}	The standard deviation of RA_3^{in}	CI4a13
	RA_3^{out}	The ratio of $\frac{RA_1^{out}}{A_p^l}$	CI4a2
	RA_{3avg}^{out}	The average value of RA_3^{out}	CI4a21
	$RA_{3min/max}^{out}$	The minimum/maximum value of RA_3^{out}	CI4a22
	RA_{3std}^{out}	The standard deviation of RA_3^{out}	CI4a23
	RA_4^{in}	The ratio of $\frac{RA_2^{in}}{A_t^{in}}$	CI4a3
	RA_{4avg}^{in}	The average value of RA_4^{in}	CI4a31
	$RA_{4min/max}^{in}$	The minimum/maximum value of RA_4^{in}	CI4a32
	RA_{4std}^{in}	The standard deviation of RA_4^{in}	CI4a33
	RA_4^{out}	The ratio of $\frac{RA_2^{out}}{A_t^{out}}$	CI4a4
	RA_{4avg}^{out}	The average value of RA_4^{out}	CI4a41
	$RA_{4min/max}^{out}$	The minimum/maximum value of RA_4^{out}	CI4a42
	RA_{4std}^{out}	The standard deviation of RA_4^{out}	CI4a43

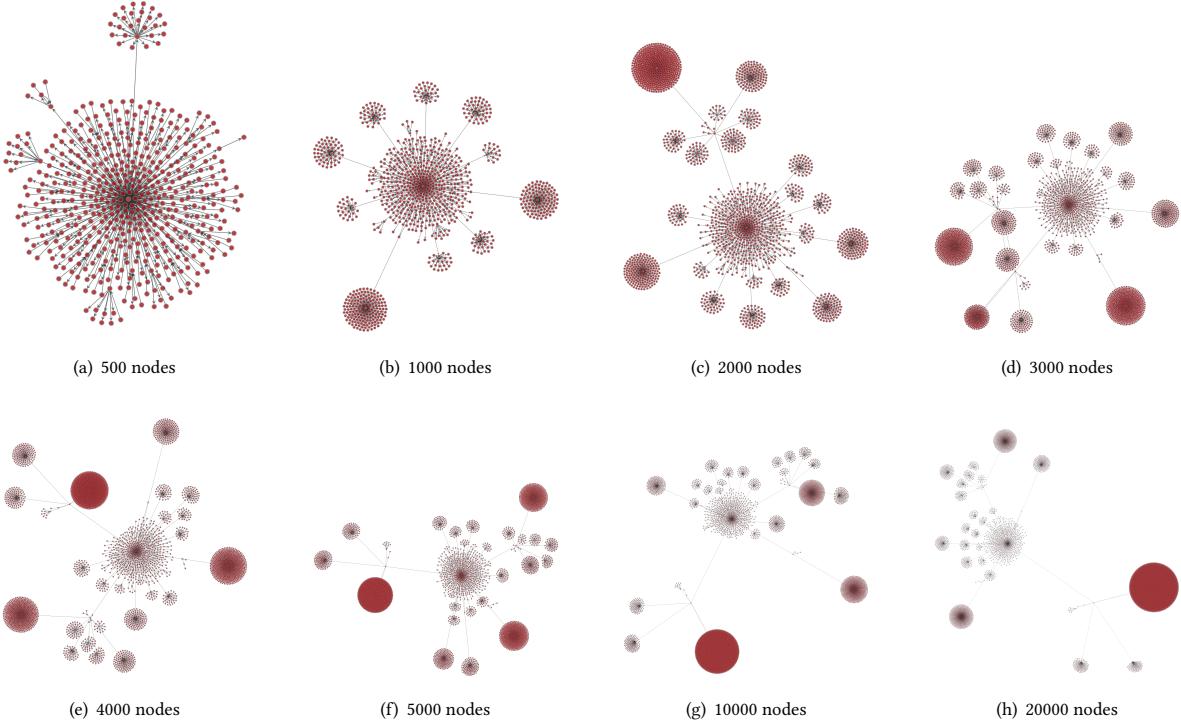


Figure 2: G_k with maximum $k = 4$ and different maximum number of nodes

- **Betweenness (S4).** $G_k^{B_i}$ is the betweenness that reflects the impact of the node in G_k [3]:

$$G_k^{B_i} = \sum_{i \neq j \neq k} \frac{P_{jk}^s(i)}{P_{jk}^s}, \quad (3)$$

where P_{jk}^s is the shortest path between $node_j$ and $node_k$, $P_{jk}^s(i)$ is the number of shortest path between $node_j$ and $node_k$ while going through $node_i$.

- **Average Path (S5).** $G_k^{\bar{P}}$ is the average distance between any two nodes in G_k :

$$G_k^{\bar{P}} = \frac{1}{(G_k^A + G_k^T)(G_k^A + G_k^T - 1)} \sum_{i \neq j} len_{ij}, \quad (4)$$

where len_{ij} is the distance between $node_i$ and $node_j$.

- **Diameter (S6).** G_k^{len} is the longest distance between any two nodes in G_k .

- **Closeness Centrality (S7).** $G_k^{Ce_i}$ measures the closeness from one node to other nodes, which can provide the efficiency of spreading information of a node [19]:

$$G_k^{Ce_i} = \frac{1}{\sum_{j=1}^{n-1} len_{min}(i, j)}, \quad (5)$$

where n is the total number of nodes in G_k , $len_{min}(i, j)$ is the minimum distance between $node_i$ and $node_j$, and $node_i$ is the node required to be measured.

- **PageRank (S8).** $G_k^{PR(i)}$ is value of PageRank of $node_i$ in G_k , which evaluates the influence of the node [20]:

$$G_k^{PR(i)} = \frac{1 - \alpha}{G_k^A + G_k^T} + \alpha \sum_{j \in \beta(i)} \frac{G_k^{PR(j)}}{\gamma(j)}, \quad (6)$$

where α is a damping factor, $\beta(i)$ are the in-neighbors of $node_i$, and $\gamma(j)$ is the out-degree of $node_j$.

- **Density (S9).** G_k^{ds} is the density of G_k that measures the density according to the edge connectivity [4]:

$$G_k^{ds} = \frac{E}{(G_k^A + G_k^T)(G_k^A + G_k^T - 1)}, \quad (7)$$

where E is the total number of edges.

5 EXPERIMENTS

5.1 Basic Settings

We mainly utilize *graph-tool* [24] to implement the framework proposed above. Also, we use *networkx* [10] to test and verify our methods and algorithms on small graphs⁹. Our device configuration is shown in Table 4.

Additionally, we mainly use *scikit-learn* [23] and *xgboost* [6] to implement common machine learning models. We choose k -nearest neighbors (KNN) algorithm, decision tree (DT), random forest (RF), multilayer perceptron (MLP), and XGBoost (XGB) to test the BABD-13.

⁹<https://github.com/Y-Xiang-hub/Bitcoin-Graph-Analysis>

Algorithm 1 G_k generation while preserving original structure

Input: the dictionary $dict$, the Bitcoin transaction graph G , the Bitcoin address Ads

Output: the k -hop subgraph G_k

```

1: function GET_MAPPED_NODE(node)
2:   if node  $\notin$  dict then
3:     dict(node)  $\leftarrow$  len(dict)
4:   end if
5:   return dict(node)
6: end function
7: function GK_GEN(G, Ads, depth)
8:   subgraph_edges  $\leftarrow$  list()
9:   Gview  $\leftarrow$  G as undirected
10:  dist(Ads)  $\leftarrow$  0
11:  for edge in Gview.edges_in_bfs_order(Ads) do
12:    f  $\leftarrow$  edge.f
13:    t  $\leftarrow$  edge.t
14:    if f  $\notin$  dict then
15:      break
16:    end if
17:    if dist(f)  $<$  depth then
18:      if t  $\notin$  dist then
19:        dist(t)  $\leftarrow$  dist(f)+1
20:      end if
21:      f'  $\leftarrow$  Get_Mapped_Node(f)
22:      t'  $\leftarrow$  Get_Mapped_Node(t)
23:      if edge(f,t)  $\in$  G.E then
24:        subgraph_edges.append(edge(f',t'))
25:      end if
26:      if edge(t,f)  $\in$  G.E then
27:        subgraph_edges.append(edge(t',f))
28:      end if
29:    end if
30:  end for
31:  return  $G_k$ 
32: end function

```

Table 4: Computer Configuration

Hardware / Software	Configuration
CPU	Intel Xeon Silver 4210R 2.40 GHz
memory	320 GB RAM
operating system	Ubuntu 21.04
python	3.9.5
graph-tool	2.42

5.2 Experiment Processes

After Bitcoin ledger (JSON files) and labeled Bitcoin both have been collected completely, our following experiment consists of three phases - transaction graph construction, features extraction and address category modeling. The specific experiment implementation is illustrated below.

First, we construct the directed heterogeneous multigraph-based Bitcoin transaction graph through JSON files of Bitcoin ledger as

Fig. 1 by graph generation function implemented by *graph-tool* (the JSON example and code can be found in our project). The point of this step is that coinbase transactions are different from normal transactions, thus, we add them to the graph in different methods respectively.

Then, we extract intuitive features in *SI* of labeled Bitcoin addresses directly from the constructed Bitcoin graph, leveraging functions implemented by *graph-tool*. However, it is more complicated and slower for extracting features from *LSI* because we need to use Algorithm 1 to generate a concrete subgraph for each labeled Bitcoin address and then obtain the features by complex network functions provided by *graph-tool* from the generated G_k .

Besides, we adopt parallel computing ways to accelerating the speed to get the features of Bitcoin addresses. Next, we store the result in a CSV file and preprocess it. There is an important step in preprocessing the raw CSV file, which is filling the missing values. To deal with it, we use the zero value to fill all the missing values in the raw CSV file according to our understanding of our proposed features on the Bitcoin transaction graph.

Finally, we use *scikit-learn* and *xgboost* to implement common machine learning models to test BABD-13¹⁰. We perform an 8:2 split of the training set and testing set on the BABD-13 using function *train_test_split()* with parameter *test_size=0.2*. Then we choose the min-max normalization to preprocess the split BABD-13 training set and testing set by function *MinMaxScaler()*. It is noted that we divide the BABD-13 into two parts that are *SI* with 132 features and *LSI* with 16 features.

The task for these models is a multiclass classification on BABD-13 and the number of samples for each class is quite imbalanced. We select 5 machine learning models to test BABD-13. The models with their corresponding methods in *scikit-learn* and parameter configurations are shown below.

- **KNN**, built by function *KNeighborsClassifier()* from *scikit-learn* with parameters *n_neighbors=4*, *algorithm='kd_tree'*, and *weights='distance'*.
- **DT**, built by function *DecisionTreeClassifier()* from *scikit-learn* with parameters *criterion='entropy'* and *splitter='best'*.
- **RF**, built by function *RandomForestClassifier()* with parameter *n_estimators=200*.
- **MLP** by function *MLPClassifier()* from *scikit-learn* with parameters *max_iter=1000*, *solver='adam'*, and *hidden_layer_sizes=(100, 100, 100)*.
- **XGB** by function *XGBClassifier()* from *xgboost* with parameters *objective='multi:softmax'*, *num_class=13*, *eval_metric='mlogloss'*, *learning_rate=0.5*.

5.3 Results and Analysis

The experiment results of different 5 machine learning models are shown in Table 5¹¹. The results of (*SI+LSI*) witness that the accuracy rate, precision, recall, and f1-score can reach at least 93.24%, 92.80%, 93.24%, and 92.97%, and at most 96.71%, 96.46%, 96.71%, and 96.57%.

¹⁰For easily reproducing our work, we set parameter *random_state=9* for every method and model if it exists

¹¹The parameter *average='weighted'* is used for calculating precision, recall, and f1-score

Table 5: The Performance of Different Machine Learning Models

Method	Evaluation Metric			
	Accuracy	Precision	Recall	F1-score
KNN ^{SI}	0.6862	0.6667	0.6862	0.6749
KNN ^{LSI}	0.9418	0.9381	0.9418	0.9397
KNN ^{SI+LSI}	0.9324	0.9280	0.9324	0.9297
DT ^{SI}	0.7061	0.6961	0.7061	0.7001
DT ^{LSI}	0.9376	0.9375	0.9376	0.9375
DT ^{SI+LSI}	0.9444	0.9447	0.9444	0.9446
RF ^{SI}	0.7566	0.7366	0.7566	0.7342
RF ^{LSI}	0.9565	0.9532	0.9565	0.9544
RF ^{SI+LSI}	0.9598	0.9564	0.9598	0.9577
MLP ^{SI}	0.7366	0.7081	0.7366	0.7063
MLP ^{LSI}	0.9331	0.9289	0.9331	0.9306
MLP ^{SI+LSI}	0.9406	0.9376	0.9406	0.9385
XGB ^{SI}	0.7825	0.7624	0.7825	0.7654
XGB ^{LSI}	0.9604	0.9579	0.9604	0.9590
XGB ^{SI+LSI}	0.9671	0.9646	0.9671	0.9657

From the results, it is also obvious that for the two kinds of indicators *LSI* and *SI*, the former one performs better than *SI* greatly. The average accuracy rates of these 5 models are 94.58% and 73.36% respectively on *LSI* and *SI*. Also, for example, on KNN, the accuracy rate improves by 25.56% after using *LSI* instead of *SI* which is the maximal increase among these models. Besides, at least the accuracy rate enhances 17.79% on XGB after applying *LSI* instead of *SI*.

However, the influence of *SI* can not be ignored because we can see that the combination of *SI+LSI* leads to the increment of all evaluation metrics to a certain extent for all the models except KNN. Although the improvement is not significant, it proves that there are valuable features in *SI* and we will further analyze them in the next section.

Additionally, from the perspective of the model level, XGB displays the best results that are all higher than 96.46% on all evaluation metrics and RF is following whose results are all higher than 95.64%. The results of evaluation metrics show that each metric of DT and MLP is close, and the differences in each metric are fewer than 0.71%. Besides, the differences in each metric for MLP and KNN are range from 0.82% to 0.96%. Besides, MLP performs better clearly when using only *SI* compared with the other two models, the accuracy rate is 73.66% and is close to RF.

According to the evaluation metrics, the effect order of these 5 models on BABD-13 is XGB>RF>DT≈MLP>KNN on BABD-13. Except for the above analysis, we also take a closer look at the relationships among these 148 features on BABD-13 using heatmap, from which we can observe the degree of relevance between every

two features, and we will use the heatmap to analyze features in BABD-13 in the following section. The generated heatmap of BABD-13 is shown in Fig. 3.

To sum up, these 5 common machine learning models show good performance on BABD-13, and it proves that BABD-13 is meaningful and helpful for further studying the behavioral patterns of different types of Bitcoin addresses. Moreover, some specific features that are listed on BABD-13 need to be considered further for the Bitcoin address behaviors study.

6 DISCUSSION

6.1 Feature Selection

We utilize two methods to measure the importance of each feature in BABD-13, one is the function *SelectKBest()* from *scikit-learn*, and another is the attribute *feature_importances_*. The first method is used for general single feature selection and the second one is applied for a specific model to choose features.

Based on the feature selection results on both ways, we find that the top 10 features are listed as follows:

- (1) The maximum out-degree in G_k (S2-2).
- (2) The standard deviation of the in-degree and out-degree in G_k (S1-6).
- (3) The standard deviation of in-degree in G_k (S1-2).
- (4) The degree correlation of G_k (S3).
- (5) $RAa_{3_{min}}^{in}$ (PAIa21-1).
- (6) A_{min}^{ti} (PTIa41-2).
- (7) The longest distance between any two nodes in G_k (S6).
- (8) The closeness centrality of G_k (S5).
- (9) $\Delta RA_{2_{max}}^{in}$ (CI3a32-2).
- (10) The density of G_k (S7).

In addition, we also rank the top 1 feature of each indicator type (i.e., *PAI*, *PDI*, *PTI*, *CI*, and *LSI*) are showed below:

- *PAI*: $RAa_{3_{min}}^{in}$ (PAIa21-1).
- *PDI*: The out-degree of an address (PDIa1-2).
- *PTI*: A_{min}^{ti} (PTIa41-2).
- *CI*: $\Delta RA_{2_{max}}^{in}$ (CI3a32-2).
- *LSI*: The maximum out-degree in G_k (S2-2).

6.2 Model Improvement by Selected Features

The final results on KNN and RF the new selected feature set are shown in Table 6. It is obvious that after the targeted selecting features in *SI* on BABD-13, the performance of KNN and RF improve. Specifically, the accuracy rate increases by 1.20% on KNN which is obvious. Additionally, the accuracy rate on RF goes up by 0.37% which is not evident compared to KNN. As for other evaluation metrics in Table 6 for KNN and RF, there are similar increments as the corresponding accuracy rate trends above.

The experimental results prove that the current feature combination, i.e., *SI+LSI* can be re-selected in *SI*, for higher model performance for different machine learning models. However, for different models, the same feature might have an unequal influence (positive or negative) that will cost time to test a number of feature combinations to obtain more results. The effect on different machine learning models due to feature selection on BABD-13 can be further

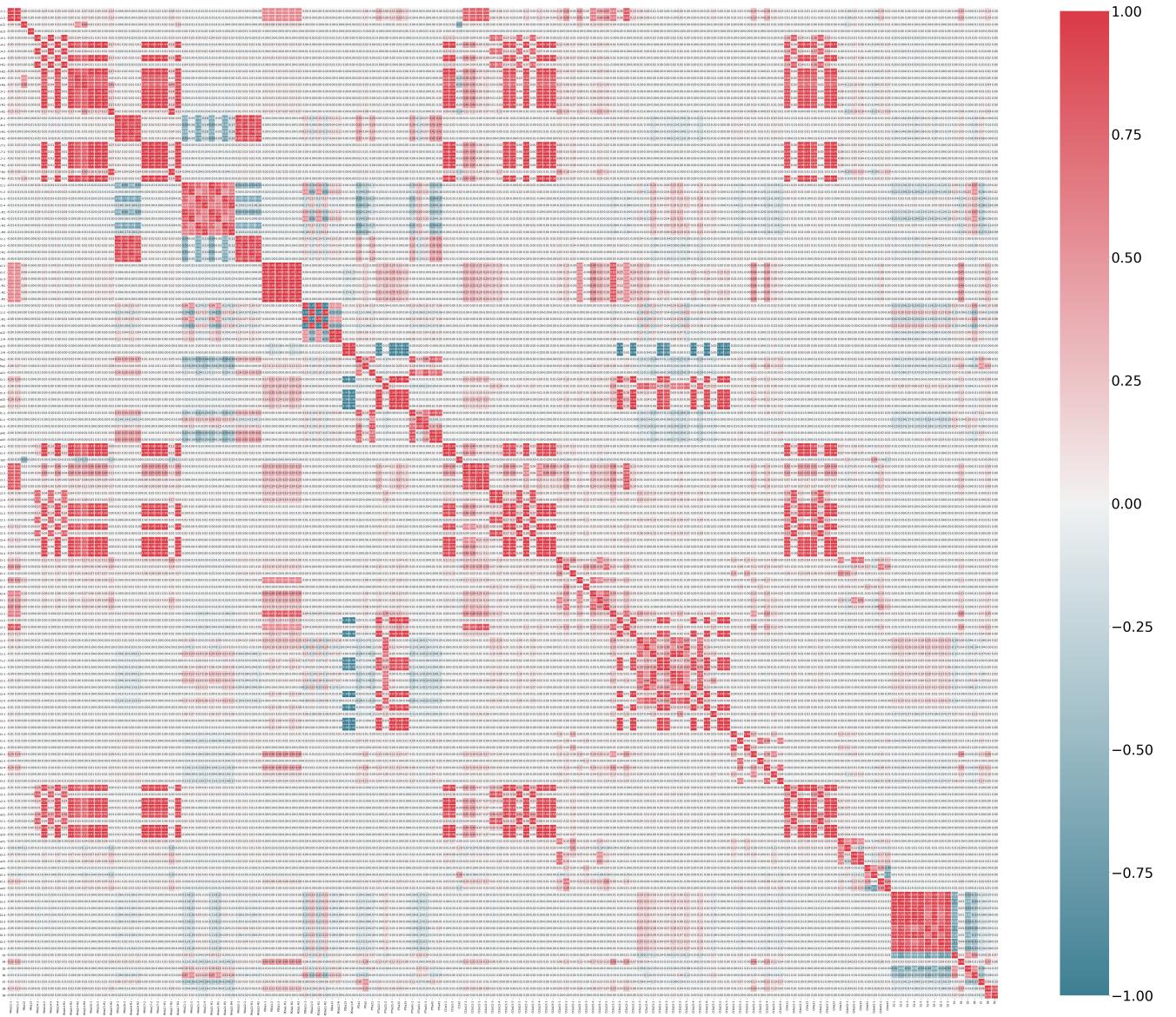


Figure 3: Bitcoin address feature heatmap

studied in future work. The experiment records are stored in our Github project.

6.3 Feature Analysis

In this part, we select the top 1 feature in each type of indicator mentioned in Section 6.1 and analyze the effect and data distribution of features in BABD-13.

In Fig. 4(a), the data distributions of all types of addresses have two peaks, furthermore, it is clear that the data distributions of all types of addresses on this feature are symmetric with different shapes and the symmetry axes are around 0.5.

For Fig. 4(b), the data distributions are similar and have one peak that is not apparent. In addition, we can see that the data

distributions of types 2, 5, 6, 8, 9 are almost the same. We think that is the reason why this feature ranks low in the feature importance compared with other features selected in this part.

In Fig. 4(c), we can observe that there are several types that are similar. For example, types 1, 3, 10, and 11. But still, other types have different distributions, where the data distribution of type 8 is the most distinct.

For Fig. 4(d), the data distribution of types 7 and 8 are unique to other types. Besides, types 0 and 1 are similar; types 1, 3, 6, and 11 are similar; types 2, 4, 5, 8, and 10 are similar. That might be the reason why this feature ranked not high in the top 10 influential features.

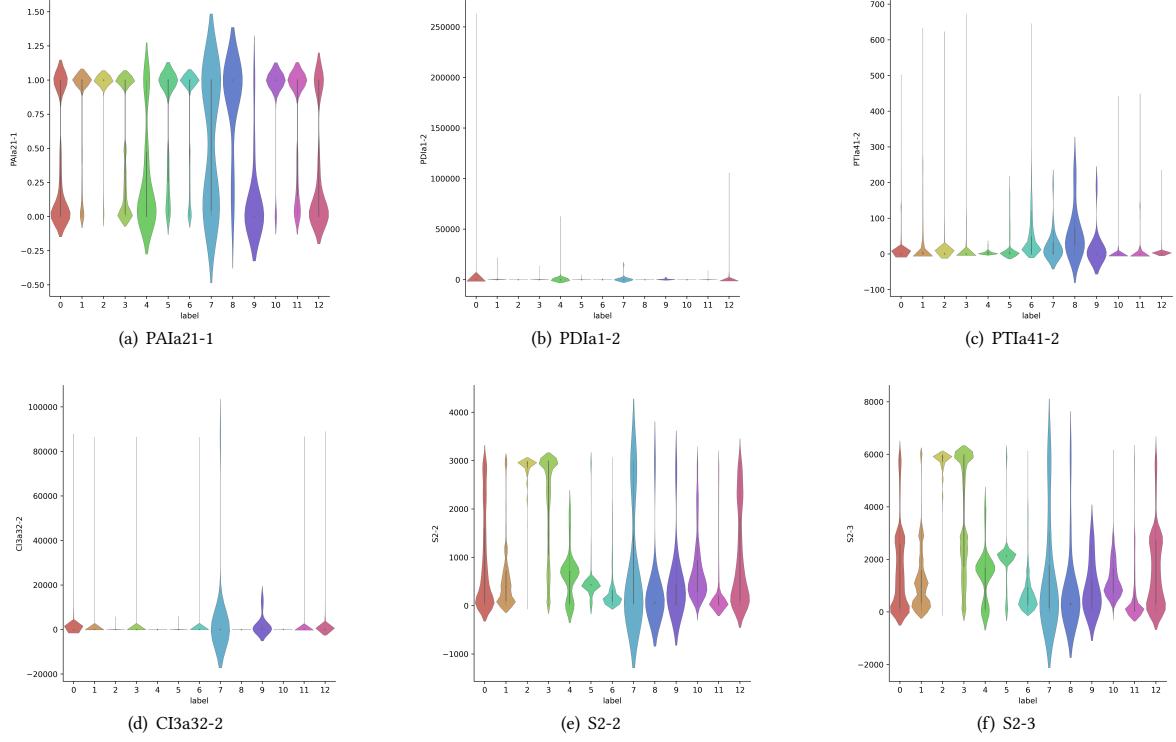


Figure 4: The violin plots of BABD-13 feature value distribution

Table 6: The Performance on KNN and RF Using Selected Features

Method	Evaluation Metric			
	Accuracy	Precision	Recall	F1-score
KNN ^{SI}	0.6862	0.6667	0.6862	0.6749
KNN ^{LSI}	0.9418	0.9381	0.9418	0.9397
KNN ^{SI+LSI}	0.9324	0.9280	0.9324	0.9297
KNN^{NEW}	0.9444	0.9409	0.9444	0.9424
RF ^{SI}	0.7566	0.7366	0.7566	0.7342
RF ^{LSI}	0.9565	0.9532	0.9565	0.9544
RF ^{SI+LSI}	0.9598	0.9564	0.9598	0.9577
RF^{NEW}	0.9635	0.9603	0.9635	0.9616

In Fig. 4(e) data distributions of all types have obvious different shapes which means it is easy to distinguish different types of addresses using this feature. Concretely, the differences show in many aspects, such as the number of peaks, the location of the peak, the shape of the distribution, and the symmetry.

Finally, we select a high correlation feature related to S2-3, i.e., S2-3, to compare with their similarities and differences. In Fig. 4(e) and Fig. 4(f), it is clear that most features with the same type have

similar shapes. There are some with partially similar shapes, such as types 0, 1, and 12. That proves the reason why the correlation is high of S2-2 and S2-3 in the heatmap.

Moreover, based on the results above, we can raise some new valuable questions. For example, how to use the feature value distributions to extract novel meaningful features? And is it possible for an address to have more than one label due to there being overlapping in specific feature value distributions on different types of addresses? These issues we will continue exploring through BABD-13 in the near future.

7 CONCLUSION AND FUTURE WORK

In this paper, we proposed a framework for Bitcoin address behavior pattern analysis on the Bitcoin transaction graph constructed by the directed heterogeneous multigraph. The latter is built based on our Bitcoin address behavior dataset (BABD-13). We also proposed a novel approach to generate k -hop subgraph for an address node from the entire Bitcoin transaction graph built by the directed heterogeneous multigraph in order to extract LSI. We then used BABD-13 to evaluate five common machine learning models, and achieved an accuracy rate of between 93.24% and 96.71%. If we wish to achieve better performance, we can also simplify BABD-13.

No work is perfect, and there are a number of potential research extensions. First, BABD-13 can be studied further in order to help us determine other useful features and characteristics. Second, we are also exploring how we can extend our proposed framework from Bitcoin address behavioral analysis to Bitcoin transaction mode

analysis, and use the combined framework to obtain a more comprehensive pattern framework or model. Finally, we also consider designing a method to identify entities, such as in the approaches outlined in [5, 15, 18, 30, 31, 40, 41], from the directed heterogeneous multigraph Bitcoin transaction graph to distinguish and analyze different entities.

ACKNOWLEDGMENTS

The research was financially supported by the CCF-NSFOCUS Kun-Peng Scientific Research Fund (No. CCF-NSFOCUS2021008), the Provincial Key Research and Development Program of Hubei (No. 2020BAB105), the National Natural Science Foundation of China (No. 61972366), the Foundation of Henan Key Laboratory of Network Cryptography Technology (No. LNCT2020-A01), and the Foundation of Hubei Key Laboratory of Intelligent Geo-Information Processing (No. KLIGIP-2021B06).

Also, we would like to thank Prof. Linchuan Xiang from the School of Physics, Huazhong University of Science and Technology, and Aleš Janda the founder of WalletExplorer.

REFERENCES

- [1] Israa Alqassem, Iyad Rahwan, and Davor Svetinovic. 2018. The anti-social system properties: Bitcoin network data analysis. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 50, 1 (2018), 21–31.
- [2] Massimo Bartoletti, Barbara Pes, and Sergio Serusi. 2018. Data mining for detecting bitcoin ponzi schemes. In *2018 Crypto Valley Conference on Blockchain Technology (CVCB)*. IEEE, 75–84.
- [3] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang. 2006. Complex networks: Structure and dynamics. *Physics Reports* 424, 4 (2006), 175–308. <https://doi.org/10.1016/j.physrep.2005.10.009>
- [4] Béla Bollobás and Bela Bollobas. 1998. *Modern graph theory*. Vol. 184. Springer Science & Business Media.
- [5] Tao-Hung Chang and Davor Svetinovic. 2018. Improving bitcoin ownership identification using transaction patterns analysis. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 50, 1 (2018), 9–20.
- [6] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (San Francisco, California, USA) (KDD '16). ACM, New York, NY, USA, 785–794. <https://doi.org/10.1145/2939672.2939785>
- [7] Mauro Conti, Ankit Gangwal, and Sushmita Ruj. 2018. On the economic significance of ransomware campaigns: A Bitcoin transactions perspective. *Computers & Security* 79 (2018), 162–189.
- [8] Damiano Di Francesco Maesa, Andrea Marino, and Laura Ricci. 2016. Uncovering the Bitcoin Blockchain: An Analysis of the Full Users Graph. In *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, 537–546. <https://doi.org/10.1109/DSAA.2016.52>
- [9] Alex Greaves and Benjamin Au. 2015. Using the bitcoin transaction graph to predict the price of bitcoin. *No Data* (2015).
- [10] Aric Hagberg, Pieter Swart, and Daniel S Chult. 2008. *Exploring network structure, dynamics, and function using NetworkX*. Report. Los Alamos National Lab.(LANL), Los Alamos, NM (United States).
- [11] Yang Li, Yue Cai, Hao Tian, Gengsheng Xue, and Zibin Zheng. [n.d.]. Identifying illicit addresses in bitcoin network. In *International Conference on Blockchain and Trustworthy Systems*. Springer, 99–111.
- [12] Kevin Liao, Ziming Zhao, Adam Doupe, and Gail-Joon Ahn. [n.d.]. Behind closed doors: measurement and analysis of CryptoLocker ransoms in Bitcoin. In *2016 APWG symposium on electronic crime research (eCrime)*. IEEE, 1–13.
- [13] Xiao Fan Liu, Xin-Jian Jiang, Si-Hao Liu, and Chi Kong Tse. 2021. Knowledge Discovery in Cryptocurrency Transactions: A Survey. *IEEE Access* 9 (2021), 37229–37254. <https://doi.org/10.1109/access.2021.3062652>
- [14] Xiao Fan Liu, Huan-Huan Ren, Si-Hao Liu, and Xian-Jian Jiang. 2021. Characterizing key agents in the cryptocurrency economy through blockchain transaction analysis. *EPJ Data Science* 10, 1 (2021), 1–13.
- [15] John V Monaco. 2015. 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, 945704.
- [16] Pranav Nerurkar, Dhiren Patel, Yann Busnel, Romaric Ludinard, Saru Kumari, and Muhammad Khurram Khan. 2021. Dissecting bitcoin blockchain: Empirical analysis of bitcoin network (2009–2020). *Journal of Network and Computer Applications* 177 (2021), 102940.
- [17] M. E. J. Newman. 2002. Assortative Mixing in Networks. *Phys. Rev. Lett.* 89 (Oct 2002), 208701, Issue 20. <https://doi.org/10.1103/PhysRevLett.89.208701>
- [18] Jonas David Nick. 2015. *Data-driven de-anonymization in bitcoin*. Thesis.
- [19] Kazuya Okamoto, Wei Chen, and Xiang-Yang Li. 2008. Ranking of closeness centrality for large-scale social networks. In *International workshop on frontiers in algorithmics*. Springer, 186–195.
- [20] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The PageRank Citation Ranking : Bringing Order to the Web. In *WWW 1999*.
- [21] Masarah Paquet-Clouston, Bernhard Haslhofer, and Benoit Dupont. 2019. Ransomware payments in the bitcoin ecosystem. *Journal of Cybersecurity* 5, 1 (2019), tyz003.
- [22] Masarah Paquet-Clouston, Matteo Romiti, Bernhard Haslhofer, and Thomas Charvat. [n.d.]. Spams meet cryptocurrencies: Sextortion in the bitcoin ecosystem. In *Proceedings of the 1st ACM conference on advances in financial technologies*. 76–88.
- [23] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.
- [24] Tiago P. Peixoto. 2014. The graph-tool python library. *figshare* (2014). <https://doi.org/10.6084/m9.figshare.1164194>
- [25] Manoj Kumar Popuri and Mehmet Hadi Gunes. 2016. *Empirical analysis of cryptocurrencies*. Springer, 281–292.
- [26] Stephen Ranshous, Cliff A Joslyn, Sean Kreyling, Kathleen Nowak, Nagiza F Samatova, Curtis L West, and Samuel Winters. [n.d.]. Exchange pattern mining in the bitcoin transaction directed hypergraph. In *International Conference on Financial Cryptography and Data Security*. Springer, 248–263.
- [27] Matteo Romiti, Aljosha Judmayer, Alexei Zamyatkin, and Bernhard Haslhofer. 2019. A deep dive into bitcoin mining pools: An empirical analysis of mining shares. *arXiv preprint arXiv:1905.05999* (2019).
- [28] Dorit Ron and Adi Shamir. 2013. Quantitative analysis of the full bitcoin transaction graph. In *International Conference on Financial Cryptography and Data Security*. Springer, 6–24.
- [29] Luca Serena, Stefano Ferretti, and Gabriele D'Angelo. 2021. Cryptocurrencies activity as a complex network: Analysis of transactions graphs. *Peer-to-Peer Networking and Applications* (2021), 1–15.
- [30] Raj Sanjay Shah, Ashutosh Bhatia, Atith Gandhi, and Shray Mathur. 2021. Bitcoin Data Analytics: Scalable techniques for transaction clustering and embedding generation. In *2021 International Conference on COMmunication Systems & NETworkS (COMSNETS)*. IEEE, 1–6.
- [31] Meng Shen, Junxian Duan, Ning Shang, and Liehuang Zhu. 2020. Transaction Deanonymization in Large-Scale Bitcoin Systems via Propagation Pattern Analysis. In *International Conference on Security and Privacy in Digital Economy*. Springer, 661–675.
- [32] Bishenghui Tao, Hong-Ning Dai, Jiajing Wu, Ivan Wang-Hei Ho, Zibin Zheng, and Chak Fong Cheang. 2022. Complex Network Analysis of the Bitcoin Transaction Network. *IEEE Transactions on Circuits and Systems II: Express Briefs* (2022).
- [33] Bishenghui Tao, Ivan Wang-Hei Ho, and Hong-Ning Dai. [n.d.]. Complex Network Analysis of the Bitcoin Blockchain Network. In *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 1–5.
- [34] Natkamon Tovanich, Nicolas Soulié, Nicolas Heulot, and Petra Isenberg. [n.d.]. An empirical analysis of pool hopping behavior in the Bitcoin blockchain. In *2021 IEEE International Conference on Blockchain and Cryptocurrency*.
- [35] Kentaroh Toyoda, P Takis Mathopoulos, and Tomoaki Ohtsuki. 2019. A novel methodology for hyip operators' bitcoin addresses identification. *IEEE Access* 7 (2019), 74835–74848.
- [36] Marie Vasek and Tyler Moore. [n.d.]. Analyzing the Bitcoin Ponzi scheme ecosystem. In *International Conference on Financial Cryptography and Data Security*. Springer, 101–112.
- [37] Mark Weber, Giacomo Domeniconi, Jie Chen, Daniel Karl I Weidle, Claudio Bellei, Tom Robinson, and Charles E Leiserson. 2019. Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics. *arXiv preprint arXiv:1908.02591* (2019).
- [38] Jiajing Wu, Jielu Liu, Weili Chen, Huawei Huang, Zibin Zheng, and Yan Zhang. 2021. Detecting mixing services via mining bitcoin transaction network with hybrid motifs. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* (2021).
- [39] Jiajing Wu, Jielu Liu, Yijing Zhao, and Zibin Zheng. 2021. Analysis of cryptocurrency transactions from a network perspective: An overview. *Journal of Network and Computer Applications* (2021), 103139.
- [40] Haaroon Yousaf, George Kappos, and Sarah Meiklejohn. 2019. Tracing transactions across cryptocurrency ledgers. In *28th USENIX Security Symposium (USENIX Security 19)*, 837–850.
- [41] Baokun Zheng, Liehuang Zhu, Meng Shen, Xiaojiang Du, and Mohsen Guizani. 2020. Identifying the vulnerabilities of bitcoin anonymous mechanism based on address clustering. *Science China Information Sciences* 63, 3 (2020), 1–15.
- [42] Francesco Zola, Jan Lukas Bruse, Maria Eguimendia, Mikel Galar, and Raul Orduna Urrutia. 2019. Bitcoin and cybersecurity: temporal dissection of blockchain

