BABD: A Bitcoin Address Behavior Dataset for Pattern Analysis

Yuexin Xiang$$^\text{a}$$, Graduate Student Member, IEEE, Yuchen Lei$$^\text{a}$$, Ding Bao$$^\text{a}$$, Tianlian Li$$^\text{a}$$, Qingqing Yang, Wenmao Liu$$^\text{a}$$, Wei Ren$$^\text{b}$$, Member, IEEE, and Kim-Kwang Raymond Choo$$^\text{c}$$, Senior Member, IEEE

Abstract—Cryptocurrencies have dramatically increased adoption in mainstream applications in various fields such as financial and online services, however, there are still a few amounts of cryptocurrency transactions that involve illicit or criminal activities. It is essential to identify and monitor addresses associated with illegal behaviors to ensure the stability of the cryptocurrency ecosystem. In this paper, we propose a framework to 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, which is the largest labeled Bitcoin address behavior dataset publicly available to our knowledge. We also propose a novel and efficient subgraph generation algorithm called BTC-SubGen 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. We then conduct 13-class classification tasks on BABD-13 by five machine learning models namely $k$-nearest neighbors algorithm, decision tree, random forest, multilayer perceptron, and XGBoost, the results show that the accuracy rates are between 93.24% and 97.13%. In addition, we study the relations and importance of the proposed features and analyze how they affect the effect of machine learning models. Finally, we conduct a preliminary analysis of the behavior patterns of different types of Bitcoin addresses using concrete features and find several meaningful and explainable modes.

Index Terms—Cryptocurrency, Bitcoin transaction, subgraph generation algorithm, machine learning, behavior pattern.

I. INTRODUCTION

CRYPTOCURRENCIES, such as Bitcoin, remain increasingly popular. For example, according to CoinMarketCap, the global cryptocurrency market capital is $1.07 Trillion U.S. dollars and Bitcoin’s market capitalization is estimated to be at $50 Trillion U.S. dollars\footnote{https://coinkapercap.com} (as of September 5, 2023). Similar to fiat currencies, there are concerns about the misuse of cryptocurrencies (e.g., cybercriminal and darknet markets). However, unlike fiat currencies, one can more easily trace most cryptocurrency transactions partly because of the transparent nature of the public ledger, except for some privacy-preserving cryptocurrencies such as Monero and Zcash.

It is, therefore, unsurprising that there have been numerous attempts to design techniques to facilitate cryptocurrency transaction tracing \cite{1, 2, 3, 4, 5, 6, 7}. For example, in the context of Bitcoin transactions, one could utilize transaction graph analysis to determine or classify different address types. There are, however, many challenges and limitations in existing approaches, ranging from information loss to incompleteness (including incompleteness of feature sets and address types in the dataset) to the framework. Another limitation we observe is the lack of a comprehensive dataset with available code that can be used as a baseline for the research community.

Therefore, we present a general framework used for building a Bitcoin transaction graph typed as the directed heterogeneous multigraph, and the generated graph can be applied for empirical, statistical, and machine learning-based analysis of behavior patterns of different types of address nodes. The main contributions of this work are shown as follows:

1556-6021 © 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
We collect Bitcoin transactions that occur between 12 July 2019 and 26 May 2021 and compile them into a dataset (i.e., BABD-13), including 13 types of 544,462 Bitcoin addresses with labels and 5 categories of indicators with 148 features, which, to our knowledge, is the largest Bitcoin address behavior dataset publicly available and can be found on Kaggle.\(^2\)

- We propose a novel and efficient algorithm called BTC-SubGen to generate \(k\)-hop subgraphs, used for extracting structural features, from the entire Bitcoin transaction graph built by the directed heterogeneous multigraph.
- We utilize five common machine learning models for the 13-class classification tasks, including \(k\)-nearest neighbors algorithm, decision tree, random forest, multilayer perceptron, and XGBoost, to demonstrate the effectiveness of the BABD-13. The accuracy rates ranging between 93.24% and 97.13% indicate the high quality and reliability of BABD-13.
- We analyze the behavior patterns of different types of Bitcoin addresses via specific features to reveal valuable insights implicit in Bitcoin transaction graphs.

In the next section, we review the recent state-of-the-art works related to Bitcoin transaction analysis. Section III introduces the essential backgrounds of this paper and Section IV proposes the general framework for Bitcoin transaction graph construction with specific indicators and algorithms. The experiment process and corresponding results are illustrated in Section V and Section VI applies the proposed dataset to a preliminary analysis of address behavior patterns. At last Section VII concludes this paper.

II. RELATED WORK

Due to the increasing popularity of cryptocurrencies, there has been a corresponding increase in concerning such currencies from governments and regulatory agencies. Bitcoin is the most famous cryptocurrency with the highest market cap currently, which reinforces the importance of understanding the Bitcoin transaction graph. The 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 through the Bitcoin transaction graph. However, to be more specific, how do we better understand and more accurately distinguish legitimate from suspicious cryptocurrency addresses by the Bitcoin transaction graph?

A. Bitcoin Transaction Graph Analysis

Many approaches to analyzing the overall characteristics of Bitcoin transaction graphs have been proposed in the literature from a network perspective. 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 [3] measured the general characteristics of the Bitcoin network using the complex network theory.

Similarly, Serena et al. [5] considered 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. In 2021, Tao et al. [6] 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.

B. Legitimate Bitcoin Address Analysis

Several studies have been conducted for different categories of address analysis via the Bitcoin transaction graph to identify specific patterns for legitimate addresses. Ranshous et al. [8] used the directed hypergraph to analyze the patterns associated with exchange addresses, and proposed different types of short thick bands to identify the patterns. In addition, they also applied basic machine learning methods to classify early exchange addresses.

Focusing on mining pools, Romiti et al. [9] selected three of four of the biggest pools to empirically investigate the relationships among them. They mainly utilized economic activity pattern-based ways to explore the relationships between miner-owned addresses and pools. In another work, Tovanich et al. [10] 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 those pool fees and payout schemes are the two most important factors to influence the behaviors of miner-owned addresses.

C. Illicit Bitcoin Address Analysis

There have been attempts to detect activities associated with illicit addresses or transactions using Bitcoin transaction graph [11], [12], [13], [14]. For instance, Liao et al. [15] measured data such as Bitcoin amount in ransomware cases and the resulting financial loss. Conti et al. [16] 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. Paquet-Clouston et al. [17] proposed an efficient framework to identify and collect ransomware attack-related Bitcoin transaction addresses. Their research demonstrated the change in the Bitcoin amount over time from 2013 to 2017.

Bartoletti et al. [18] 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 [19] obtained their Ponzi scheme data from a bitcoin forum\(^3\) and observed that cybercriminals interacted more frequently with the victims and posted more comments in the thread. In addition, Toyoda et al. [20]

\(^{2}\)https://www.kaggle.com/datasets/lemonx/babd13

\(^{3}\)https://bitcointalk.org
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 historical records and applied supervised learning to classify the HYIP addresses. The accuracy rate of their approach is as high as 93.75%.

D. Challenges in Current Approaches

Plenty of approaches are designed to study the similarities and differences of features and patterns among distinct kinds of Bitcoin addresses or transactions within the Bitcoin transaction graph [2], [11], [21], [22], [23], [24]. However, most existing approaches mentioned previously prefer to utilize the simplified Bitcoin transaction graph (e.g., undirected simple graph), rather than its original structure, in their analysis [12]. Consequently, this may result in significant information loss. Meanwhile, the categories of Bitcoin addresses in most existing works are inaccurate and/or incomplete, like at most seven types of addresses being analyzed together in recent works of Lin et al. [25]. This is insufficient to have an in-depth understanding of the behavior patterns of different types of addresses in the Bitcoin transaction graph.

Additionally, the indicators utilized for analysis proposed in previous works, such as those of [2], [22], [23], and [24], are generally not systematic and/or comprehensive. To be more specific, most of these works ignore many essential indicators that can be extracted or calculated from the Bitcoin transaction graph. They also fail to classify indicators to offer deeper insights into the importance of different kinds of feature sets. It is also noted that these works do not take account into features, especially structural features, extracted from local structures of subgraphs. Besides, most of these mentioned methods are difficult to replicate since there is no clear framework or process for how the Bitcoin transaction graph is built and generated without available datasets or codes.

III. BACKGROUND MATERIALS

A. Bitcoin Transaction 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 [21], [26]. For a general analysis of the Bitcoin network, address (Ads) nodes or transaction (Tx) nodes are the main objects [11], [18], [22], [27], especially Tx nodes because they include more information. Besides, there are also a great number of works that regard related addresses as an entity for analysis [23], [28].

However, from our perspective, the above and similar approaches to simplify the Bitcoin transaction graph will lead to information loss in the Bitcoin graph analysis. In this paper, to diminish 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 Di Francesco Maesa et al. [29], which includes characteristics of both the Ads node and the Tx node to ensure the accuracy of the analysis.

Concretely, Ads nodes indicate Bitcoin addresses that can launch transactions including sending and receiving Bitcoins and Tx nodes represent transactions that occur among Ads nodes that contain concrete transaction information. Our designed structure is shown in Fig. 1, which explains the specific information contained in Ads nodes, Tx nodes, and edges.

B. Behavior Classification and Definition

It is significant to classify and define typical behaviors of Bitcoin addresses exactly. Specifically, the concepts and basic motivations of different illicit and licit Bitcoin addresses will help us improve the effect of classification tasks for Bitcoin addresses and deeper understand the patterns that exist in Bitcoin address behaviors. We will study and analyze the mainstream 13 types of illicit and legitimate Bitcoin addresses in this paper, which are listed as follows:

1) **Blackmail.** Cryptocurrency blackmail has three typical categories that are ransomware, sextortion, and scam. Apart from the blackmail methods, these blackmail types are similar in most aspects, which utilize some ways to threaten 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 only conducted by cryptocurrency. Examples are bond markets and P2P lending platforms.

6) **P2P Financial Service.** P2P financial applications reward users with cryptocurrency for their contributions. These incentives encourage users to complete tasks beneficial to the organizers, such as increasing the number of advertisement clicks. This kind of service includes faucets, video sharing, and affiliate marketers.

7) **Gambling.** Cryptocurrency gambling denotes playing casino games, such as blackjack and roulette, in which only cryptocurrencies can be utilized as wagers.

8) **Government Criminal Blocklist.** Government criminal blocklist includes cryptocurrency addresses highly suspected or confirmed involved in criminal activities according to different countries’ laws.

9) **Money Laundering.** Money laundering involves transferring dirty cryptocurrency from illicit transactions to legitimate addresses. These transactions are carried out through complicated methods that are difficult 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 interest by cryptocurrency to the former investors through the investments from currently involved investors. Therefore, the investors will believe that the initiator can easily earn profits for them in a relatively short time.

11) **Mining Pool.** Cryptocurrency mining pools consist of a large number of miners who contribute their computational resources together to improve 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 wallets owned by ordinary people who use cryptocurrency as a payment way in daily life, such as shopping and dining.

### C. Data Collection

There are two kinds of data we need to collect 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 API. Our current research on 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. Due to the limitation of our device performance, we use only 100,001 blocks to build the Bitcoin transaction graph. However, our approach is capable of building Bitcoin transaction graphs of any size.

**Bitcoin Address:** Compared with the Bitcoin ledger, it is harder to collect Bitcoin addresses with labels. In order to obtain more data while keeping the accuracy of the data, we divide our collected data into two types, i.e., the strong address (SA) and the 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 dataset and government blocklists. Furthermore, we also find valuable SA data from the highly influential papers, we find valuable SA data from the highly influential papers[16], [18]. Besides, WAs are gathered from the public reported Bitcoin address dataset and a number of social accounts on Twitter. All the addresses with labels collected are listed in Table I.

### IV. Proposed Scheme

Referring to the indicators and classification methods presented in a variety of state-of-the-art works [2], [21], [22], [23], [24], [30], [31], [32], [33], [34], [35], [36], we proposed a novel framework for extracting and analyzing Bitcoin address behaviors from the Bitcoin transaction graph that is 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.

### A. Statistical Indicator

SI in our proposed scheme is divided into four concrete indicator types that are introduced as follows:

| 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 Blocklist     | 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) |

3https://www.walletexplorer.com
4https://btc.com
5https://www.walletexplorer.com
6https://home.treasury.gov
7https://www.bitcoinabuse.com
8https://twitter.com
• **Pure Amount Indicator (PAI).** PAI is related to the token amount attributes of Ads nodes in the Bitcoin transaction graph, for example, the total input token amount of an address node.

• **Pure Degree Indicator (PDI).** PDI includes the degree-related attributes of Ads nodes in the Bitcoin transaction graph, such as the out-degree of an address node.

• **Pure Time Indicator (PTI).** PTI is time-related attributes of Ads nodes, including the lifecycle of an address node and the active period of an address node.

• **Combination Indicator (CI).** CI is the combinations of the features from PAI, PDI, and PTI, for instance, the total in-degree for each day within the active days.

The basic SI is shown in Table II that includes PAI, PDI, and PTI, and features from CI are illustrated in Table III. In these two tables, there are a total of 132 features (the corresponding explanation of each notation is available on GitHub	extsuperscript{9}) and we need to further explain three points. 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. To be more specific, we recompute the same features after merging the edges with the same direction in the Bitcoin transaction graph with their attributes and make them newly added features (e.g., recomputing PAIa11-1 as PAIa11-R1 using the Bitcoin transaction graph by merging the edges with the same direction).

### B. Local Structural Indicator

In the Bitcoin network, Tx nodes link Ads nodes to form a graph, thus we think the features extracted from a subgraph comprising of these two types of nodes that are close to an address can provide local structural information of the address, which is capable of contributing to address type identification.

Unlike the usual $k$-hop subgraphs, we consider containing nodes that are close to each other in the corresponding undirected graph, while reflecting the actual structure of the local Bitcoin network. Therefore, we propose the **BTC-SubGen** as shown in Algorithm 1 to generate $k$-hop subgraph $G_k$ for each labeled Ads node and then obtain useful LSIs from generated subgraphs.

In Algorithm 1, we regard the original graph as an undirected graph and traverse each edge in breadth-first order starting from a given address. After that, the nodes linked 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, to limit the size of $G_k$, we set two thresholds for the proposed algorithm. One threshold is the maximum value of $k$ and we select $k = 4$ as the maximum value. Another threshold is the maximum number of Ads nodes and Tx nodes and we choose 3,000 as the maximum number of the sum of Ads node and Tx node.

There are two primary reasons why we choose the above two hyperparameters:

1. 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 time cost increases dramatically when hyperparameters exceed the limits, especially when $k > 4$.

2. From a local structure perspective, when $k = 4$ we take Fig. 2 as an example, there are eight figures of $G_k$ with different numbers of nodes generated from the same Ads.

Our experimental results from various $k$-hop subgraphs reveal that Fig. 2(a), Fig. 2(b), and Fig. 2(c) contain less structural information than the other five figures of $G_k$. This observation aligns with the intuitive visual insights apparent in these figures. Besides, Fig. 2(g) and Fig. 2(h) have too much information that is similar to the entire transaction graph and cannot easily get valuable distinctive local structural information. Compared among the rest Fig. 2(d), Fig. 2(e), and Fig. 2(f) we observe that their structural are very similar, thus we choose $G_k$ with 3,000 nodes as the hyperparameter, i.e., Fig. 2(d).

---

	extsuperscript{9}https://github.com/Y-Xiang-hub/Bitcoin-Address-Behavior-Analysis

	extsuperscript{10}1BtcBoSSnqe8mFJCUEyCNmo3EcF8Yzhpnc from Ponzi.
We also need to clarify that we find similar conclusions as shown in Fig. 2 in plenty of other address nodes with different types of Bitcoin addresses during a number of experiments, herein is just an instance.
We have to admit that these two hyperparameters might not be optimal but the experimental results in the following section show that the current \textsl{LSI} features obtained from generated subgraphs are enough for training common machine learning models to achieve high performance. The method for \textsl{LSI} features extraction from the subgraphs generated by \textsc{BTC-SubGen} are shown below, and the explanations of a total of 16 \textsl{LSI} features are available on our GitHub:

- **Average Degree (S1).** $G^7_k$ means the average degree \cite{37} of the nodes in $G_k$:

$$G^7_k = \frac{1}{G^5_k + G^7_k \left( \sum_{i=1} G^5_i + \sum_{j=1} G^7_j \right)}, \quad (1)$$

**TABLE III**

| Indicator | Notation | Description | Dataset |
|-----------|----------|-------------|---------|
| **PAI + PDI** | RA\textsubscript{in/out}\textsuperscript{1}, RA\textsubscript{2} | The ratio of $A^\text{in}_{A_k}$ and $A^\text{out}_{A_k}$ | C1a1 |
| | RA\textsubscript{3}, RA\textsubscript{4} | The ratio of $D_{A_k}/D_{A_k}$ | C1a2 |
| **PAI + PTI** | $A^\text{in/out}_{A_k}$ | The input/output token amount of an address in each basic unit of the active period | C1a1 |
| | $A_{\text{avg}}^\text{in/out}$ | The average input/output token amount of an address in each basic unit of the active period | C1a1 |
| | $A_{\text{min/max}}^\text{in/out}$ | The minimum/maximum input/output token amount of an address in each basic unit of the active period | C1a1 |
| | RA\textsubscript{in/out}\textsuperscript{2} | The ratio of $A^\text{in}_{A_k}$ and $A^\text{out}_{A_k}$ | C1a2 |
| | RA\textsubscript{2,avg} | The average value of $RA_{A_k}^{\text{in/out}}$ | C1a2 |
| | RA\textsubscript{2,min/max} | The minimum/maximum value of $RA_{A_k}^{\text{in/out}}$ | C1a2 |
| | RA\textsubscript{2,std} | The standard deviation of $RA_{A_k}^{\text{in/out}}$ | C1a2 |
| | $\Delta A^\text{in/out}\textsubscript{A_{k,all}}$ | The change of $A_{A_{k,all}}$ | C1a4 |
| | $\Delta RA_{A_{k,all}}^{\text{in/out}}$ | The ratio of $\Delta A^\text{in}_{A_k}/A^\text{in}_{A_k}$ and $\Delta A^\text{out}_{A_k}/A^\text{out}_{A_k}$ | C1a4 |
| | $\Delta RA_{A_{k,all}}^{\text{avg}}$ | The average value of $\Delta RA_{A_k}^{\text{in/out}}$ | C1a4 |
| | $\Delta RA_{A_{k,all}}^{\text{min/max}}$ | The minimum/maximum value of $\Delta RA_{A_k}^{\text{in/out}}$ | C1a4 |
| | $\Delta RA_{A_{k,all}}^{\text{std}}$ | The standard deviation of $\Delta RA_{A_k}^{\text{in/out}}$ | C1a4 |
| **PDI + PTI** | $A^\text{in/out}_{A_k}$ | The in-degree/out-degree of an address in each basic unit of the active period | C1a1 |
| | $A_{\text{avg}}^\text{in/out}$ | The average in-degree/out-degree of an address in each basic unit of the active period | C1a1 |
| | $A_{\text{min/max}}^\text{in/out}$ | The minimum/maximum in-degree/out-degree of an address in each basic unit of the active period | C1a1 |
| | RA\textsubscript{3,avg} | The average value of $RA_{A_k}^{\text{in/out}}$ | C1a2 |
| | $\text{RA}_{A_{k,all}}^{\text{in/out}}$ | The minimum/maximum value of $\text{RA}_{A_k}^{\text{in/out}}$ | C1a2 |
| | RA\textsubscript{3,avg} | The standard deviation of $\text{RA}_{A_k}^{\text{in/out}}$ | C1a2 |
| | $\Delta RA_{A_{k,all}}^{\text{in/out}}$ | The ratio of $\Delta A^\text{in}_{A_k}/A^\text{in}_{A_k}$ and $\Delta A^\text{out}_{A_k}/A^\text{out}_{A_k}$ | C1a3 |
| | $\Delta RA_{A_{k,all}}^{\text{avg}}$ | The average value of $\Delta RA_{A_k}^{\text{in/out}}$ | C1a3 |
| | $\Delta RA_{A_{k,all}}^{\text{min/max}}$ | The minimum/maximum value of $\Delta RA_{A_k}^{\text{in/out}}$ | C1a3 |
| | $\Delta RA_{A_{k,all}}^{\text{std}}$ | The standard deviation of $\Delta RA_{A_k}^{\text{in/out}}$ | C1a3 |
| **PAI + PDI** | RA\textsubscript{in} | The ratio of $A^\text{in}_{A_k}$ | C1a4 |
| + PTI | RA\textsubscript{in,avg} | The average value of $RA_{A_k}^{\text{in}}$ | C1a4 |
| | RA\textsubscript{in,min/max} | The minimum/maximum value of $RA_{A_k}^{\text{in}}$ | C1a4 |
| | RA\textsubscript{in,std} | The standard deviation of $RA_{A_k}^{\text{in}}$ | C1a4 |
| | RA\textsubscript{out} | The ratio of $A^\text{out}_{A_k}$ | C1a4 |
| | RA\textsubscript{out,avg} | The average value of $RA_{A_k}^{\text{out}}$ | C1a4 |
| | RA\textsubscript{out,min/max} | The minimum/maximum value of $RA_{A_k}^{\text{out}}$ | C1a4 |
| | RA\textsubscript{out,std} | The standard deviation of $RA_{A_k}^{\text{out}}$ | C1a4 |
| | RA\textsubscript{4} | The ratio of $\Delta RA_{A_k}^{\text{out}}$ | C1a4 |
| | RA\textsubscript{4,avg} | The average value of $RA_{A_k}^{\text{out}}$ | C1a4 |
| | RA\textsubscript{4,min/max} | The minimum/maximum value of $RA_{A_k}^{\text{out}}$ | C1a4 |
| | RA\textsubscript{4,std} | The standard deviation of $RA_{A_k}^{\text{out}}$ | C1a4 |

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
where $G_k^A$ is the number of $Ads$ nodes, $G_k^I$ is the number of $Tx$ nodes, $A_k$ is the degree of the $Ad$ 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 $LSI$.

- **Degree Distribution (S2).** $G_k^{d(d)}$ is the degree distribution [37] that means the proportion of the node with the $d$ degree in $G_k$ [37]. 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 [37] 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 [38] is selected to measure $G_k$:

$$G_k^{C_{d}} = \frac{1}{\sum_{e_{ij}} d_i d_j - \left(\frac{1}{\sum_{e_{ij}} d_i^2 + d_j^2}\right)\left(\frac{1}{\sum_{e_{ij}} d_i^2 + d_j^2}\right)^2} ,$$

(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.

- **Betweenness (S4).** $G_k^B$ is the betweenness that reflects the impact of the node in $G_k$ [37]:

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

(3)

where $P_{jk}^s$ is the number of 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).** $G_k^P$ is the average distance between any two nodes in $G_k$:

$$G_k^P = \frac{1}{(G_k^A + G_k^I)(G_k^A + G_k^I - 1)} \sum_{i \neq j} \text{len}_{ij},$$

(4)

where $\text{len}_{ij}$ 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).** $G_k^{C_{e}}$ measures the closeness from one node to other nodes, which can provide the efficiency of spreading information of a node [39]:

$$G_k^{C_{e}} = \frac{1}{\sum_{j=1}^{n-1} \text{len}_{\text{min}}(i, j)},$$

(5)

where $n$ is the total number of nodes in $G_k$, $\text{len}_{\text{min}}(i, j)$ 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 [40]:

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

(6)

where $\alpha$ 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).** $G_k^{d_{s}}$ is the density of $G_k$ that measures the density according to the edge connectivity [41]:

$$G_k^{d_{s}} = \frac{E}{(G_k^A + G_k^I)(G_k^A + G_k^I - 1)},$$

(7)

where $E$ is the total number of edges.

V. EXPERIMENT AND RESULT ANALYSIS

A. Basic Setting

We mainly utilize graph-tool [42] on Linux to implement the framework proposed above. We also apply networkx [43] to test and verify our methods and algorithms on small graphs. Our device configuration is shown in Table IV and the code is available on our GitHub.

In addition, we use scikit-learn [44] and xgboost [45] 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 conduct 13-class classification tasks on BADB-13.

B. Experiment Process

After the Bitcoin ledger (JSON files) and labeled Bitcoin addresses 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 extracted from the Bitcoin ledger as Fig. 1 by graph generation function implemented by graph-tool (the example of JSON file and code for graph construction is available on our GitHub). The point of this step is that coinbase transactions are different from general 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 to extract structural features from $LSI$ because we need to use BTC-SubGen 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 $G_k$.

Notably, we adopt parallel computing ways to accelerate 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 in 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 knowledge of our proposed features on the Bitcoin transaction graph.

Finally, we use scikit-learn and xgboost to implement common machine learning models\(^\text{11}\) to execute 13-class classification tasks on BABD-13 (the first 132 features of BABD-13 belong to SI, and the next 16 features are LSI). However, the number of samples for each class is relatively imbalanced, which leads to a great challenge for the classification task.

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().

Specifically, we select five machine learning models to conduct 13-class classification tasks on BABD-13. The models with their corresponding methods in scikit-learn and parameter configurations are shown as follows.

- **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.

### C. Experimental Result and Analysis

The experimental results for the 13-class classification tasks of five distinct machine learning models are shown in Table V,\(^\text{12}\) where SI+LSI represent all features in BABD-13 that are used for machine learning model training, and single SI or LSI indicates only its involved features are applied for the training. The results of usage of entire features in BABD-13 witness that the accuracy rate, precision, recall, and f1-score can reach a minimum of 93.24%, 92.80%, 93.24%, and 92.97%, respectively, and a maximum of 96.71%, 96.46%, 96.71%, and 96.57%, respectively.

From the results, it is clear that for the two kinds of indicators LSI and SI, LSI performs better than SI greatly. The average accuracy rates of these five models are 94.58% and 73.36% respectively on LSI and SI. Noteworthy, 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 cannot 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 following 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 follows 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%. The differences in each metric for MLP and KNN 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% which is close to RF.

According to the evaluation metrics, the effect order on the 13-class classification tasks of five models on BABD-13 is XGB>RF>DT≈MLP> KNN on BABD-13. Beyond the previous analysis, we examine the relationships among these 148 features on BABD-13 using a heatmap as shown in Fig. 3. Based on the feature heatmap, we can specifically observe the degree of relevance between each pair of features. Fig. 3 illustrates that the majority of the features contained in BABD-13 are low or no relevance to each other. However, there are still several highly relevant features, especially within CI, which can be further studied and enhanced in future work.

In summary, these five common machine learning models show outstanding performance on 13-class classification tasks using BABD-13, supporting the high quality and reliability of the dataset. This makes BABD-13 valuable for further studying the behavioral patterns of different types of Bitcoin addresses. In addition, to highlight the distinctiveness and state-of-the-art aspects of the BABD-13, we compared it with other similar influential datasets, and the results are demonstrated in Table VI.

---

11 Random_state=9 is set for each method or model if it exists.
12 Average=’weighted’ is used for calculating precision, recall, and f1-score.
TABLE VI

| Item            | Scheme       | BARD          | Weber et al. [13] | Li et al. [11] | Ranshous et al. [8] | Michalski et al. [46] | Lin et al. [25] |
|-----------------|--------------|---------------|-------------------|---------------|---------------------|-----------------------|------------------|
| **Time Range**  |              |               |                   |               |                     |                        |                  |
|                 |              |               |                   |               | Sep. 29, 2011 ~    | May 2, 2018 ~         | Jan. 3, 2009 ~     |
|                 |              |               |                   |               | Apr. 22, 2015      | May 3, 2018           |                  |
| **Graph Size**  |              | 516,167,131 nodes | 203,769 nodes     | N/A           | N/A                 | N/A                   | N/A              |
|                 |              | 713,703,239 edges | 234,353 edges     |               |                     |                       |                  |
| **Sample Size** |              | 544,462       | 46,564            | 1,234,047     | 972,866             | 8,808                 | 26,313           |
| **Object**      |              | Ads node      | Tx node           | Ads node      | Ads node            | Ads node              | Ads node         |
| **Feature Size**|              | 148           | 166               | 92            | N/A                 | 149                   | 64               |
| **Number of Types** | 13 | 2             | 2                 | 2             | 6                   | 7                     |
| **Model**       |              | DT, KNN, MLP, RF, XGB | GCN, LR, MLP, RF | ANN, RF, SVM, XGB | AdaBoost, LR, MLP, RF, SVM | DT, ET, NN, RF, SVM | AdaBoost, LightGB, LR, MLP, NN, RF, SVM |
| **Performance** |              | Accuracy: 97.1% | Precision: 97.1% | Recall: 97.1% | F1-score: 97.0%     | Precision: 93.5%     | Recall: 99.7%     |
|                 |              | Precision: 96.8% | F1-score: 96.0%   |               |                     | Precision: 99.7%     | F1-score: 99.7%   |
| **Dataset Availability** | ✓ | ✓ | ✓ | x | ✓ | x |
| **Code Availability** | ✓ | ✓ | ✓ | x | x | x |

*Performance denotes best model performance, but since the listed works focus on diverse classification tasks, the comparison is provided for reference.*

3) Calculate the between-class variance ($B$) by computing the weighted average of the squared differences between class means and the overall mean.

4) Compute the $F$-value as the ratio of between-class variance to within-class variance:

$$F = \frac{B}{W}$$

The $F$-value indicates the significance of the feature’s contribution to the classification. Therefore, we adopt the function `SelectKBest()` with the `f_classif` scoring function from scikit-learn to select the features. `f_classif` uses the above method to compute the $F$-value for each feature. Then `SelectKBest()` selects the most useful features which have the highest $F$-value. As a result, 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) The ratio of the minimum input token amount of an address node to the total input token amount of an address node ($RA_{3 \text{min}}$, PAIa21-1).
6) The minimum transaction time interval of an address node ($A_{3 \text{min}}^t$, PTIa41-2).
7) The longest distance between any two nodes in $G_k$ (S6).
8) The closeness centrality of $G_k$ (S5).
9) The maximum value of the ratio of the change in in-degree to each transaction time interval for the address node in chronological order ($\Delta RA_{3 \text{min}}^t$, CI3a32-2).
10) The density of $G_k$ (S7).
Additionally, we rank the top 1 feature of each category of indicator (i.e., \(\text{PAI}^1\), \(\text{PDI}^1\), \(\text{PTI}^1\), \(\text{CI}^1\), and \(\text{LSI}^1\)) which are shown below:

- \(\text{PAI}^1\): The ratio of the minimum input token amount of an address node to the total input token amount of an address node \((R_{\text{Ai}}^{\text{min}}, \text{PAI}_{21-1})\).
- \(\text{PDI}^1\): The out-degree of an address \((\text{PDI}_{1-2})\).
- \(\text{PTI}^1\): The minimum transaction time interval of an address node \((\text{PTI}_{41-2})\).
- \(\text{CI}^1\): The maximum value of the ratio of the change in degree to each transaction time interval for the address node in chronological order \((\Delta \text{RA}^{\text{min}}, \text{CI}_{3a32-2})\).
- \(\text{LSI}^1\): The maximum out-degree in \(G_k\) (S2-2).

According to the feature relevance and importance results in the previous parts, we reselect a new set of features (i.e., \(\text{new}\) in Table VII) from BABD-13 for 13-class classification tasks. The final results on KNN and RF the newly selected feature set are shown in Table VII. After applying the new feature set in \(SI\) on BABD-13, the performances of KNN and RF improve. Specifically, the accuracy rate increases by 1.20\% on KNN which is obvious. The accuracy rates on RF and XGB go up by 0.37\% and 0.42\% which are not evident compared to KNN. However, the XGB reaches the new highest level of accuracy rate of 97.13\% after utilizing crafted chosen features. As for other evaluation metrics in Table VII for KNN and RF, there are similar increments as the corresponding accuracy rate trends above.

The new experimental results illustrate that the current feature set, i.e., \(SI+LSI\), can be reselected for higher model performance targeting different machine learning models. However, for various models, the same feature might have an unequal influence (positive or negative) that will take time to test a number of subsets of \(SI+LSI\) to obtain more outcomes (that is also the reason why we only choose these three models as examples). The effect on machine learning models due to different feature subsets of BABD-13 can be further studied in future work. The experiment records of the new feature set are available on our GitHub.

### E. Feature Analysis

In this subsection, we select the top 1 feature in each type of indicator mentioned in Section V-D and analyze the effect and data distribution of features in BABD-13. It is also noted that types 0-12 respectively represent blackmail, cyber-security service, darknet market, centralized exchange, P2P financial infrastructure service, P2P financial service, gambling, government criminal blacklist, money laundering, Ponzi scheme, mining pool, tumbler, and individual wallet.

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, which is meaningful for address type classification.

For Fig. 4(b), the data distributions are similar and have one peak that is not apparent, and the data distributions of types 2, 5, 6, 8, and 9 are almost the same. We think that is one of the reasons 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 but not identical, such as types 1, 3, 10, and 11. However, other types have different distributions, where the data distribution of type 8 is the most distinct. This provides enough information to distinguish different types of addresses.

For Fig. 4(d), the data distributions of types 7 and 8 are unique to other types. Besides, distributions of types 0 and 12 are similar. Likewise, types 1, 3, 6, and 11 demonstrate similar distributions, as do types 2, 4, 5, 8, and 10. As a result, the similarities of distributions in various types of this feature may be the reason why it ranks not high in the top influential features.

In Fig. 4(e), data distributions of all types have obviously different shapes which means it is simple 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.

In addition, we select a high correlation feature related to S2-2, i.e., the maximum total degree of the subgraph (S2-3), to compare 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. This demonstrates the cause for the correlation is high in S2-2 and S2-3 in the heatmap.

Though some Bitcoin address types present unique distributions on the specific feature (as shown in Fig. 4(c)) providing a clear basis for address category classification, other types show that relatively overlapping patterns exist. This illustrates the complexity of obtaining information on certain address classifications. Additionally, similarities between features S2-2 and S2-3 (as displayed in Fig. 4(e) and Fig. 4(f)) demonstrate the distributions of some pairs of features are highly relative and are also reflected on the feature heatmap (as presented in Fig. 3). Overall, these insights highlight the multifaceted nature of the Bitcoin network and the challenges of pattern mining.

Moreover, based on the results above, some new valuable questions can be raised. For example, how to use the feature value distributions to extract novel meaningful features? Is it possible for an address to have more than one label due to there being overlapping in specific feature value distributions on

### Table VII

| Model and Indicator | Evaluation Metric |
|---------------------|-------------------|
|                     | Accuracy | Precision | Recall | F1-score |
| KNN\(^{SI+LSI}\)    | 0.9324   | 0.9280    | 0.9324 | 0.9297   |
| KNN\(^{new}\)      | 0.9444   | 0.9409    | 0.9444 | 0.9424   |
| RF\(^{SI+LSI}\)    | 0.9598   | 0.9564    | 0.9598 | 0.9577   |
| RF\(^{new}\)       | 0.9635   | 0.9603    | 0.9635 | 0.9616   |
| XGB\(^{SI+LSI}\)   | 0.9671   | 0.9646    | 0.9671 | 0.9657   |
| XGB\(^{new}\)      | 0.9713   | 0.9693    | 0.9713 | 0.9702   |

**TABLE VII**

THE PERFORMANCE ON KNN, RF, AND XGB USING NEW SUBSET
different types of addresses? These questions could be further investigated through BABD-13 in future work.

F. Framework Enhancement and Application

As illustrated in Fig. 4(b), compared with other indicator categories, the differentiated information of various address types we can obtain from even an optimal feature in PDI is insufficient, which means features of PDI are inadequate for Bitcoin address 13-class classification tasks. The reason might be that the final status of features of PDI is not that important for Bitcoin address transactions and PDI features are only the first layer of Bitcoin addresses that do not include the other layers.

To solve the above problem, we consider analyzing the regular growth pattern of Bitcoin addresses in the same time interval and extracting information from these patterns as new features to enhance PDI. The key issue of this study is how to choose the time interval and how to construct suitable subgraphs from an evolution perspective. We have several initial conclusions that illustrate the feasibility of this approach in our recent work [48] which remains subsequent investigation.

Furthermore, we have applied the proposed framework to conduct an in-depth analysis of Bitcoin address subgraphs [49], where the meanings of the structural features are analyzed by combining the definitions of various address types, and the graph neural network is utilized to perform multi-class classification tasks via subgraphs extracted from the whole Bitcoin transaction graph. Besides, we have developed a novel visualization system for monitoring illicit transactions in the Bitcoin network [50] based on the proposed framework.

Thus, there are numerous valuable extended research that can be implemented for Bitcoin transaction networks based on the framework proposed in this paper, such as analyzing differences between subgraphs and the whole transaction graph with their significations and exploring the optimal hyperparameters for subgraph generation.

| Type               | AVG | MAX | MIN | MED |
|--------------------|-----|-----|-----|-----|
| Darknet Market     | 2.1 | 14  | 1   | 2   |
| Money Laundering   | 14  | 4   | 2   | 2   |
| Ponzi              | 606.4 | 3,939 | 2   | 74  |

VI. PRELIMINARY ANALYSIS OF BEHAVIOR PATTERN

A. Illicit Address Mode Analysis via PDI

Although PDI features contribute insignificantly compared with other kinds of features such as LSI in classification tasks to machine learning models based on the results of feature importance, that does not mean this kind of feature is worthless.

To explain this point, we select the total degrees of an Ads node (i.e. PDIa1-3) in PDI as an example to show how we apply features in BABD-13 and the definitions of different Bitcoin address types to analyze the differences and similarities of address behavior patterns of three illicit types of addresses, i.e., the darknet market, money laundering, and Ponzi. At first, we calculate the average, maximum, minimum, and median values of PDIa1-3 of these three types of illicit addresses, and the results are shown in Table VIII.

The organizer of Ponzi requires a steady stream of new participants to invest Bitcoins through one or a set of addresses owned by the organizer to complete the commitment for the profits of preceding participants. Thus, the average, median, and maximum values of PDIa1-3 in Ponzi are relatively high, which is generally reflected in the large number of incoming/outgoing edges on the Bitcoin transaction graph of the Ponzi address node.

In contrast, the usage frequency of darknet market and money laundering is much lower for a single address based on maximum and minimum values of PDIa1-3 in Table VIII. As for darknet market, it is because the transactions that happen in the darknet market are few due to the high threshold,
high risk, and high cost as well as the sellers in the darknet market often change Bitcoin addresses for safety concerns.

In terms of money laundering, the entity that intends to launder sensitive Bitcoin demands to hide the movements of the Bitcoin flows, then it separates the total Bitcoin into several parts and transfers them plenty of times through lots of one-time use addresses finally to one/a set of new address(es). Therefore, the addresses involved in money laundering are only being used several times (maximum four times as shown in Table VIII).

In brief, as can be seen in Table VIII combined with the definitions of these address types, the various values of PDA1-3 in Ponzi are higher than in the darknet market and money laundering which is meaningful and explainable. This also illustrates that PDA1-3 may be effective for binary-class classification tasks (i.e., distinguishing Ponzi and darknet market/money laundering). Besides, to obtain clearer characteristics of these addresses, PAI can be added to the analysis since the token amounts of single transactions involved in darknet market and money laundering are generally higher than those in most Ponzi schemes.

B. Address Pattern Analysis via LSI

LSI, as the most influential feature category among all five feature sets, merits a deeper examination. We have chosen the betweenness centrality (i.e., S4) in LSI as a case to further analyze. Concisely, the higher S4 of an Ads node in the Bitcoin subgraph represents the more Bitcoin flows transit the node, which means the more important position of this node in the subgraph. The same as in the previous section, we calculate the average, maximum, minimum, and median values of six types of addresses as presented in Table IX.

Herein, we mainly take account into the average values that reflect more on the overall condition of different address types. From Table IX, compared with other types, blackmail, P2P financial infrastructure service, Ponzi, and individual wallet have higher average values. The possible reason for high average values on S4 of P2P financial infrastructure service and individual wallet are similar in that those kinds of addresses provide some services including lending, where unrelated entities send/receive Bitcoin to these same addresses to complete contracts.

Likewise, Ponzi also involves plenty of Bitcoin sending/receiving processes as presented in the previous section, while blackmail addresses receive Bitcoin from many victims by executing successful attacks, and most of the other nodes (i.e., participants) in subgraphs are irrelevant to each other.

As for the low average values of S4 on money laundering and tumbler, it is not complicated to explain. The purpose of both money laundering and tumbler is to hide the sources or flows of sensitive Bitcoin, therefore, the addresses used for these activities should remain less frequently used. In other words, only several Bitcoin flows pass through these nodes and the tumbler or money laundering network consists of numerous nodes of this kind.

Besides, compared with tumbler, money laundering is especially sensitive as we can see from Table IX that the maximum and minimum values of money laundering are zero while the maximum value of tumbler is 0.286. This implies that there is no money laundering address with an important position in subgraphs according to S4, but the maximum value of tumbler might stand for some famous tumbler agent institutions.

VII. CONCLUSION AND FUTURE WORK

In this paper, we propose a framework for Bitcoin address behavior pattern analysis in the Bitcoin transaction graph constructed by the directed heterogeneous multigraph including SI and LSI features. Based on this framework, we build a Bitcoin address behavior dataset, i.e., BABD-13, and we propose a novel algorithm called BTC-SubGen to generate k-hop subgraph for each address node with the label from the Bitcoin transaction graph. We then utilize five common machine learning models for 13-class classification tasks on BABD-13 to evaluate the dataset and achieve accuracy rates of between 93.24% and 97.13%. We also explore and analyze the relationships among pairs of features and value distributions of features of different address types. In addition, several significant Bitcoin address behavior modes in the transaction graph are found via BABD-13.

No work is perfect, and there are a number of potential research extensions. First, BABD-13 can be studied further, especially feature selection and extension and specific pattern mining. Second, the hyperparameter choice of the subgraph generation algorithm BTC-SubGen can be optimized. Third, except for address nodes, the current framework is capable of being used for transaction node identification and analysis. Finally, we consider designing or applying a method, such as in the approaches outlined in [22], [23], [27], [28], to identify entities corresponding to addresses with labels from entity-oriented Bitcoin transaction graphs typed as the directed heterogeneous multigraph.

ACKNOWLEDGMENT

The authors would also like to thank Prof. Linchuan Xiang from the School of Physics, Huazhong University of Science and Technology; Prof. Tianqing Zhu from the School of Computer Science, University of Technology Sydney; and Aleš Janda the Founder of WalletExplorer.

REFERENCES

[1] I. Alqassem, I. Rahwan, and D. Svetinovic, “The anti-social system properties: Bitcoin network data analysis,” IEEE Trans. Syst., Man, Cybern., Syst., vol. 50, no. 1, pp. 21–31, Jan. 2020.
[2] P. Nerurkar, D. Patel, Y. Busnel, R. Ludinard, S. Kumari, and M. K. Khan, “Dissecting Bitcoin blockchain: Empirical analysis of Bitcoin network (2009–2020),” J. Netw. Comput. Appl., vol. 177, Mar. 2021, Art. no. 102940.

[3] M. K. Popuri and M. H. Gunes, Empirical Analysis of Crypto Currencies. Berlin, Germany: Springer, 2016, pp. 281–292.

[4] D. Ron and A. Shamir, “Quantitative analysis of the full Bitcoin transaction graph,” in Proc. Int. Conf. Financial Cryptography Data Secur. Cham, Switzerland: Springer, 2013, pp. 6–24.

[5] L. Serena, S. Ferretti, and G. D’Angelo, “Cryptocurrencies activity as a complex network: Analysis of transactions graphs,” Peer-Peer Netw. Appl., vol. 15, no. 2, pp. 839–853, Mar. 2022.

[6] B. Tao, I. W. Ho, and H.-N. Dai, “Complex network analysis of the Bitcoin blockchain network,” in Proc. IEEE Int. Sysmp. Circuits Syst. (ISCAS), May 2020, pp. 1–5.

[7] B. Tao, H.-N. Dai, I. W. Ho, Z. Zheng, and C. F. Cheang, “Complex network analysis of the Bitcoin transaction network,” IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 69, no. 3, pp. 1009–1013, Mar. 2022.

[8] S. Ranshous et al., “Exchange pattern mining in the Bitcoin transaction directed hypergraph,” in Financial Cryptography and Data Security. Cham, Switzerland: Springer, 2017, pp. 253–270.

[9] M. Romiti, A. Judmayer, A. Zamyatin, and B. Hashhofer, “A deep dive into Bitcoin mining pools: An empirical analysis of mining shares,” 2019, arXiv:1905.05999.

[10] N. Tovanich, N. Soulé, N. Heulot, and P. Isenberg, “An empirical analysis of pool hopping behavior in the Bitcoin blockchain,” in Proc. IEEE Int. Conf. Blockchain Cryptocurrency, May 2021, pp. 1–9.

[11] Y. Li, Y. Cai, H. Tian, G. Xue, and Z. Zheng, “Identifying illicit address classification based on transaction history summarization,” in Proc. IEEE Int. Conf. Financial Cryptography Data Secur. Cham, Switzerland: Springer, 2013, pp. 6–24.

[12] K. Liao, Z. Zhao, A. Doupe, and G.-J. Ahn, “Behind closed doors: Exposure of Bitcoin activity on the Dark Web,” J. Cybersecurity, vol. 5, no. 2, pp. 1957–1962, Dec. 2022.

[13] M. Weber et al., “Anti-money laundering in Bitcoin: Experimenting with graph convolutional networks for financial forensics,” 2019, vol. 7, pp. 74835–74848, 2019.

[14] J. Wu, J. Liu, W. Chen, H. Huang, Z. Zheng, and Y. Zhang, “Detecting mixing services via mining Bitcoin transaction network with hybrid motifs,” IEEE Trans. Syst., Man, Cybern., Syst., vol. 52, no. 4, pp. 2237–2249, Apr. 2022.

[15] K. Liao, Z. Zhao, A. Doupe, and G.-J. Ahn, “Behind closed doors: Measurement and analysis of CryptoLocker ransomware in Bitcoin,” in Proc. APWG Symp. Electron. Crime Res. (eCrime), Jun. 2016, pp. 1–13.

[16] M. Conti, A. Gangwal, and S. Ruj, “On the economic significance of ransomware campaigns: A Bitcoin transactions perspective,” Comput. Secur., vol. 79, pp. 162–189, Nov. 2018.

[17] M. Paquet-Clouston, B. Hashhofer, and B. Dupont, “Ransomware payments in the Bitcoin ecosystem,” J. Cryptography, vol. 3, no. 1, Jan. 2019, Art. no. tcy003.

[18] M. Bartoletti, B. Pes, and S. Serusi, “Data mining for detecting Bitcoin Ponzi schemes,” in Proc. 10th Conf. Blockchain Technol. (CVCBT), Jun. 2018, pp. 75–84.

[19] M. Vasek and T. Moore, “Analyzing the Bitcoin Ponzi scheme ecosystem,” in Proc. Int. Conf. Financial Cryptography Data Secur. Cham, Switzerland: Springer, 2019, pp. 101–112.

[20] B. Zheng, L. Zhu, M. Shen, X. Du, and M. Guizani, “Identifying the vulnerabilities of Bitcoin anonymous mechanism based on address clustering,” Sci. China Inf. Sci., vol. 63, no. 3, pp. 1–5, Mar. 2020.

[21] M. Vasek and T. Moore, “Analyzing the Bitcoin Ponzi scheme ecosystem,” in Proc. 28th USENIX Secur. Symp., Jul. 2019, pp. 9–20.

[22] F. Zola, J. L. Bruse, M. Eguimendia, M. Galar, and R. Orudna Urrutia, “Bitcoin and cybersecurity: Temporal dissection of blockchain data to unveil changes in entity behavioral patterns,” Appl. Sci., vol. 9, no. 23, p. 5003, Nov. 2019.

[23] Y. Li, Y. Cai, H. Tian, G. Xue, and Z. Zheng, “Identifying illicit address classification based on transaction history summarization,” in Proc. IEEE Int. Conf. Blockchain Cryptocurrency (ICBC), May 2019, pp. 302–310.

[24] K. Toyoda, P. Takis Mathiopoulos, and T. Ohtsuki, “A novel methodol-
Yuexin Xiang (Graduate Student Member, IEEE) received the B.Eng. and M.Eng. degrees in information security from the School of Computer Science, China University of Geosciences. He is currently pursuing the Ph.D. degree with the Department of Software Systems and Cybersecurity, Faculty of Information Technology, Monash University, Melbourne, Australia. His research interests include blockchain, security, and artificial intelligence.

Yuchen Lei received the B.Eng. degree in information security from the School of Computer Science, China University of Geosciences, China. He is currently pursuing the M.Eng. degree with the School of Cyber Science and Engineering, Wuhan University, China. His research interests include blockchain, edge computing, and artificial intelligence.

Ding Bao received the B.Eng. degree in artificial intelligence from the School of Computer Science, Wuhan Institute of Technology, China. He is currently pursuing the M.Eng. degree with the School of Computer Science, China University of Geosciences, China. His research interests include blockchain and artificial intelligence.

Tiantian Li received the B.Eng. degree in information security from the School of Computer Science, China University of Geosciences, and the M.Eng. degree in information technology from the Faculty of Engineering and Information Technology, The University of Melbourne, Melbourne, Australia. Her research interests include artificial intelligence, blockchain, and cybersecurity.

Qingqing Yang received the B.Eng. degree in information security from the School of Computer Science, China University of Geosciences, China, where she is currently pursuing the M.Eng. degree. Her research interests include blockchain and data mining.

Wenmao Liu received the Ph.D. degree in information security from the Harbin Institute of Technology in 2013. Then, he was a Researcher with NSFOCUS Inc. During his first two years at NSFOCUS Inc., he was also working at Tsinghua University as a Post-Doctoral Researcher. He is currently the Director of the Innovation Center; the Leader of the XingYun Laboratory, NSFOCUS Inc.; and the Co-Chair of Cloud Security Service WG, CSA. His research interests include cloud security, the IoT security, data-driven analytics, and other new research areas of network security.

Wei Ren (Member, IEEE) received the Ph.D. degree in computer science from the Huazhong University of Science and Technology, Wuhan, China, in 2006. Since 2013, he has been a Full Professor with the School of Computer Science, China University of Geosciences, Wuhan. He was with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, IL, USA, from 2007 to 2008; the School of Computer Science, University of Nevada at Las Vegas, Las Vegas, NV, USA, from 2006 to 2007; and the Department of Computer Science, The Hong Kong University of Science and Technology, Hong Kong, from 2004 to 2005. He has authored or coauthored more than 150 refereed articles, one monograph, and four textbooks. He is a Distinguished Member of the China Computer Federation. He was a recipient of 20 patents and five innovation awards.

Kim-Kwang Raymond Choo (Senior Member, IEEE) received the Ph.D. degree in information security from the Queensland University of Technology, Brisbane, QLD, Australia, in 2006. He currently holds the Cloud Technology Endowed Professorship with The University of Texas at San Antonio (UTSA), San Antonio, TX, USA. He is the Founding Co-Editor-in-Chief of ACM Distributed Ledger Technologies: Research and Practice, and Founding Chair of IEEE Technology and Engineering Management Society Technical Committee on Blockchain and Distributed Ledger Technologies.