Computer Modeling in Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2023.043469

ARTICLE

A Bitcoin Address Multi-Classification Mechanism Based on Bipartite Graph-Based Maximization Consensus

Lejun Zhang[1,2,3,*], Junjie Zhang[1], Kentaroh Toyoda[4], Yuan Liu[2], Jing Qiu[2], Zhihong Tian[2] and Ran Guo[5]

[1]College of Information Engineering, Yangzhou University, Yangzhou, 225127, China

[2]Cyberspace Institute Advanced Technology, Guangzhou University, Guangzhou, 510006, China

[3]Research and Development Center for E-Learning, Ministry of Education, Beijing, 100039, China

[4]Institute of High Performance Computing (IHPC), Agency for Science, Technology and Research (A∗STAR), Singapore, 138632, Singapore

[5]School of Physics and Materials Science, Guangzhou University, Guangzhou, 510006, China

*Corresponding Author: Lejun Zhang. Email: zhanglejun@gzhu.edu.cn

Received: 03 July 2023 Accepted: 28 September 2023 Published: 30 December 2023

ABSTRACT

Bitcoin is widely used as the most classic electronic currency for various electronic services such as exchanges, gambling, marketplaces, and also scams such as high-yield investment projects. Identifying the services operated by a Bitcoin address can help determine the risk level of that address and build an alert model accordingly. Feature engineering can also be used to flesh out labeled addresses and to analyze the current state of Bitcoin in a small way. In this paper, we address the problem of identifying multiple classes of Bitcoin services, and for the poor classification of individual addresses that do not have significant features, we propose a Bitcoin address identification scheme based on joint multi-model prediction using the mapping relationship between addresses and entities. The innovation of the method is to (1) Extract as many valuable features as possible when an address is given to facilitate the multi-class service identification task. (2) Unlike the general supervised model approach, this paper proposes a joint prediction scheme for multiple learners based on address-entity mapping relationships. Specifically, after obtaining the overall features, the address classification and entity clustering tasks are performed separately, and the results are subjected to graph-based maximization consensus. The final result is made to baseline the individual address classification results while satisfying the constraint of having similarly behaving entities as far as possible. By testing and evaluating over 26,000 Bitcoin addresses, our feature extraction method captures more useful features. In addition, the combined multi-learner model obtained results that exceeded the baseline classifier reaching an accuracy of 77.4%.

KEYWORDS

Bitcoin; multi-service classification; graph maximization consensus; data security

1 Introduction

In recent years, Bitcoin [1] has gained popularity worldwide as the most valuable cryptographic digital currency currently on the market. Fig. 1 shows the price change of Bitcoin from its launch to the present day. As Fig. 1 shows, the price of Bitcoin has not changed significantly for a long time since its launch, however, with the widespread use of blockchain and the emergence of various e-services, electronic currencies are gradually gaining respect and the price of Bitcoin has peaked in the past few years.

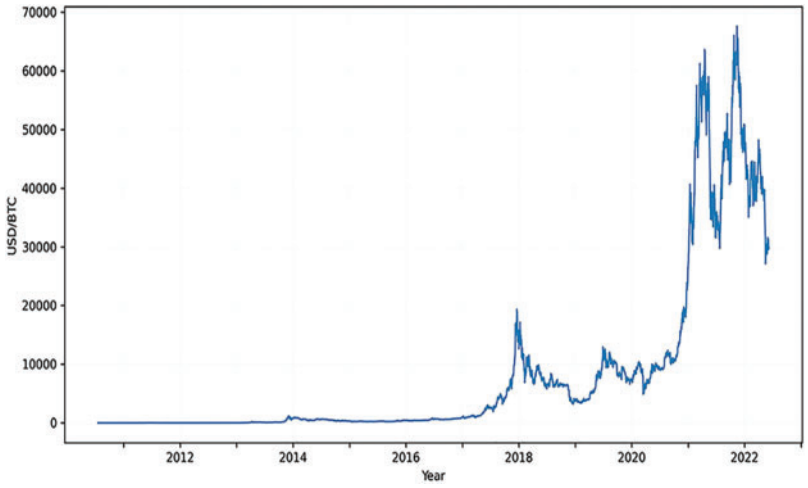

Figure 1: USD/BTC exchange rates (Source: Blockchain.info)

Tasca et al. [2] have analyzed the evolution of the Bitcoin economy and summarized three distinguishable economic regimes as the Bitcoin economy has developed and matured. Bitcoin's early phase was dominated by mining and proof-of-concept without much substantial economic activity. Followed by a period of criminal growth, when early participants were attracted to the unique properties of cryptocurrencies, hence the proliferation of "criminal" enterprises (HYIP, dark markets, etc.) under this phase. The third stage is led by legitimate exchanges, which are businesses that convert digital currencies into fiat currency to cover costs and avoid price fluctuations. They found that different commercial categories populate the various stages of Bitcoin's development and that each commercial category has its different transaction flow patterns.

Bitcoin has so many commercial categories because it has properties that traditional currencies do not have. Bitcoin has a decentralized nature, it is not controlled by governments or financial institutions, and it can even be traded directly across borders. Besides, openness, transparency, immutability and anonymity are its key characteristics. Everyone can see all the transactions that have taken place, which makes Bitcoin transactions open and transparent. Blockchain utilizes cryptography and consensus mechanisms so that all blocks retain the hash of their previous block, and if one block is changed, then all subsequent blocks must be changed, so that transaction information cannot be tampered with. The addresses involved in Bitcoin transactions are obtained as pseudonyms through layers of cryptographic calculations with the user's public key and do not contain any identifiers to verify their identity, so the user has a certain degree of anonymity. While all these features are guaranteed to improve blockchain security, the fact is that transactions on the blockchain are not secure. Zhang et al. [3] investigated that only a small percentage of blockchain platforms can achieve a set of security goals in practice. Xu et al. [4] argued that blockchain can prevent only a portion of

fraudulent and malicious activities and remains vulnerable to attacks. They suggested appropriate defensive measures and called for further research to combat malicious activities associated with blockchain.

Recently, the website blockchain.info [5] classified the existing mainstream Bitcoin cryptocurrency services in terms of risk levels with respect to their risk factors, as shown in Fig. 2. Due to the lack of regulation and investigation by law enforcement, many services come with a certain amount of risk. For example, with high-risk exchanges and coin-mixing companies, sometimes we have no way of knowing whether they are providing a legitimate service or engaging in illegal fundraising, while lost virtual currency property is difficult to retrieve without credentials. In traditional anti-fraud or anti-money laundering efforts, the government monitors the behavior of suspicious people through financial services institutions that must verify the identity of customers before providing financial services, at which point the government can accurately track down suspicious people based on perfect knowledge of the user's identity. In contrast, Bitcoin's users are difficult to identify, with no real identity information embedded in any participant's address, and worse, new addresses can be generated at will, which increases user anonymity and allows illegal activities and financial crimes to flourish, making it difficult to guarantee the security of legal user transactions. To secure user transactions, Möser et al. [6] have attempted to create a blacklisting policy for dangerous addresses in Bitcoin to enable transaction recipients to identify low-reputation addresses to avoid risk as much as possible, but the technique relies too heavily on transparent and authoritative tagging data, which may be accompanied by implications such as disinformation, information explosion and difficulty in recovering reputation, and it is difficult to implement in the anonymous Bitcoin network of flood propagation. Therefore, it is advisable to use the behavioural features of the target address to complete the identification of its possible identity prior to the transaction in order to avoid transaction risks and help the blockchain ecosystem grow in a healthy way.

Figure 2: The types and risk levels of cryptocurrency services (Source: Blockchain.info)

The rest of the paper is structured as follows. Section 2 will briefly introduce the work related to address clustering and address classification in Bitcoin, as well as the innovation points of this paper. Section 3 will introduce the general steps and specifics of the model. Section 4 will describe the experimental process of this paper, organize and evaluate the experimental results, and Section 5 will conclude the whole paper.

2 Related Work

This section first introduces the fundamentals of trading, then leads to related work on address clustering and address classification, pointing out the problems with existing research as well as stating the motivation and innovation of this paper's proposal.

2.1 Trading Fundamentals

Bitcoin is a decentralised cryptocurrency that works in a P2P (peer-to-peer) network [1]. Fig. 3 shows two examples of Bitcoin transactions. As shown in Fig. 3, Bitcoins are transferred between Bitcoin addresses via a message format called a transaction. For the second Bitcoin transaction shown in this figure, the sender and receiver of Bitcoins are identified as the input and output, respectively. The private and public keys are generated by a pair of elliptic curve encryption algorithms, but it is not possible to work backwards through the public key to get the private key. The purpose of the public key is to encrypt a message using one's own private key when dealing with the other side, who then decrypts it using their own public key to obtain the original message, this process is commonly known as a signature. As the public key is too long to use in a transaction, an algorithmic encryption of the public key hash is performed to generate the address. When a user creates a transaction to send a certain number of Bitcoins to a specific Bitcoin address, a message signature that can be calculated from their paired private keys is required. In Bitcoin, the private key signature is used to verify that the identity of the person sending the message is the person transferring the money in the transaction, and the message is confirmed if the message is decrypted.

The transaction is sent to the P2P network and checked for validity, e.g., whether the input to the transaction has not been previously spent and the additional signature is verified by the participating nodes. When a transaction is determined to be valid, it will be agreed by every participant in Bitcoin via flood propagation and stored permanently in the Bitcoin blockchain. In Bitcoin, a set of approved transactions is stored in a block in the form of a Merkle tree, and newly created blocks are periodically distributed among all nodes of the P2P network. However, due to its decentralised nature, Bitcoin lacks a trusted body to manage blocks. If a block is abandoned, the bitcoins already spent may be reused, resulting in what is known as "double spend". In fact, Bitcoin avoids double spending by rewarding bitcoins to rational nodes as an incentive. In Bitcoin, blocks are created by solving an intractable but easily checked computational puzzle. More specifically, nodes commonly referred to as miners need to lower a specified target value in the result of a (SHA-256) hash as well as the set of references to the previous block and unapproved transactions. The first miner to identify such a random number can acquire newly minted Bitcoins through a so-called coinbase transaction, shown in the first transaction in Fig. 4, and all transaction fees are included in the block. This leads to a growing chain of blocks so called the blockchain, as the previous block is needed to create the next block. It is worth noting that valid transactions agreed by the nodes as part of the block creation process are also stored permanently in the block and are not modified in any way.

Due to its credible advantages, blockchain has a wide range of applications in many areas, especially in data storage. For example, Tian et al. [7] proposed a secure digital evidence framework based on block-chain (Block-DEF) with a loosely coupled structure that allows evidence and evidentiary information to be maintained separately, alleviating the tension between evidence traceability and privacy. Wang et al. [8] addressed the serious centralization problem in DNS architecture and management by improving the consensus algorithm in blockchain to implement a blockchain-based DNS storage and retrieval system with fast consensus and low traffic. Su et al. [9] implemented a

decentralized name resolution system for IoT based on blockchain architecture to ensure that IoT names cannot be maliciously tampered with.

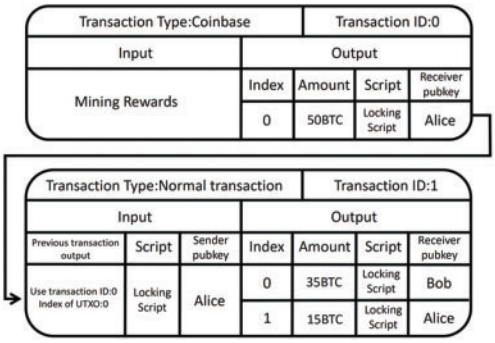

Figure 3: Details of an example transaction

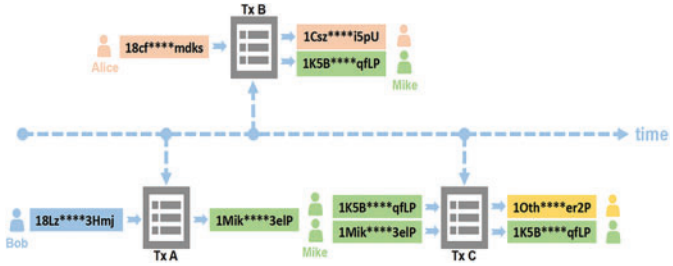

Figure 4: An example of address clustering

2.2 Bitcoin Address Clustering

In the Bitcoin system, any entity can manage more than one Bitcoin address. The de-anonymization of addresses is a key issue in analyzing addresses and a focus on the detection of illegal activities such as anti-money laundering. This technology not only helps with the task of illegal entity detection, but also with the tracking of accomplice addresses of certain illegal addresses. Several deanonymization techniques, often called address clustering algorithms, have emerged. Androulaki et al. [10] showed that despite not knowing the true identity of the owner: there are two heuristics to associate a Bitcoin address with its owner, the *Multi−input transactions heuristic* and the "*shadow*" *addresses heuristic*, respectively. The first heuristic states that all transaction addresses in a transaction belong to the same user entity. The second heuristic is that only when the output address is two if one of the addresses appears in the blockchain for the first time, it is a change address automatically generated by the system for the user and therefore belongs to the same entity as all the input addresses. Fig. 4 shows an example of address clustering, where transaction *B* and transaction *C* correspond to the "*shadow*" *addresses heuristic* and the *Multi−input transactions heuristic*, respectively.

Initially, these two address clustering methods were able to cope with most bitcoin address analysis efforts, but the consequent creation of coin mixer has somewhat impacted this approach [11]. The main idea of the mixed coin service is that multiple users transfer coins to the same transaction at the same time, rendering the multi-input heuristic rule ineffective. Some research on anti-mixed coins has also been carried out, Wu et al. [12] analyzed mixed coin transactions in Bitcoin, and they found that there are two main types of mixed coin transactions, exchange, and obfuscation, and designed heuristic rules to detect mixed coin transactions. In summary, the main limitation in identifying the controlling entity behind Bitcoin addresses is that they do not involve any identifying information linking them to the owner, and even if multiple addresses were known to be controlled by one person or one entity, it would still be unable to determine their true identity. Nevertheless, address clustering techniques can often bring a different analysis from a more macro perspective, e.g., Meiklejohn et al. [13] studied the usage of Bitcoin transactions through address clustering-based transaction graph analysis, and they showed that different clustered entities correspond to various services such as mining pools, wallets, exchanges, gambling, and money laundering, among others. For the identification work of multiple services of Bitcoin addresses, obtaining some feature information of the entity where the address is

located helps to enrich the features of the address and thus improve the classification. As the "*shadow*" *addresses heuristic* is more likely to compose too large clusters and suffer cluster crashes. In this paper we only use the *Multi−input transactions heuristic* for address clustering to obtain the entity where the address is located and other addresses of that entity.

2.3 Address Service Classification

Learning behavioural features on Bitcoin addresses to train machine learning models and identify the security factor of the target address before the transaction is completed is a reliable solution to avoid risk and secure transactions. Several studies have been conducted on the service category identification problem, which can be broadly divided into two directions, classification of addresses and classification of entities (using heuristic clustering), respectively. They usually start with feature extraction of the dataset addresses or entities, and subsequently, put the features and labeled data into the classifier for model training and result validation. For example, Akcora et al. [14] proposed a novel and effective prediction framework for ransom transactions that can automatically predict emerging ransom transactions in a limited amount of relevant historical transaction data. Jourdan et al. [15] defined some new features related to the characteristics of entities on the Bitcoin blockchain and studied their efficacy in practice. Toyoda et al. [16] summarized the historical transactions of addresses and computed the features to train the model from both address and entity perspectives, concluding that the classification results were better with entities as samples, possibly due to the greater randomness of selecting individual addresses, which made the classification results lower. Unlike Toyoda's extracting features separately for addresses and entities, Zola et al. [17] used cascaded machine learning to characterize entities, and they used the classification results of addresses within entities to vote and enriched the features of entities with the voting results as new features so that the classification results of entities followed the results of most of the addresses within them to achieve good classification results. Kanemura et al. [18] proposed a voting-based approach to identify addresses in darknet marketplaces, where all addresses within an entity are classified as darknet addresses if the addresses above a given threshold are detected to be consistent with darknet transactions, and experimental results show that the voting-based approach outperforms the non-voting-based approach in all aspects.

There is indeed cross information between the address and the entity in which it is located. The methods based on the idea of voting that have been studied do improve the results of classifying entities, however, classifying all addresses within an entity is too computationally intensive. Also, transactions are often sent to individual addresses rather than entities, and the analysis of individual addresses is the main basis for prompting transactions to be completed. So we endeavoured to solve the following question: Can the entity information of this address be used to correct the classification results of a single address in order to improve the classification accuracy of a single address? A voting strategy may not solve this problem because entities may participate in multiple services and it cannot simply be determined that all addresses within an entity participate in the same service, and the voting policy consumes a lot of computing resources. To solve the above problems, we were inspired by the article by Gao et al. [19] which combines heterogeneous source information to correct the original classification results. This paper proposes a consensus method for maximizing the consensus of address and entity classification results. Firstly, we ensure high prediction accuracy by extracting a large number of address features, secondly, we cluster highly similar entities using a small number of features of the entity where the address is located so that entities exhibiting similarity appears in the same clusters, and finally, we perform consensus on the classification results and clustering results at the output level to counteract the randomness of individual address features and optimize the classification results of individual addresses to improve the accuracy.

2.4 Motivation and Contribution

Yang et al. [20] used a Gaussian distribution unsupervised clustering method to cluster Bitcoin entities into six clusters, representing six groups of users exhibiting different features. It is not rigorous to use unsupervised clustering alone to obtain class labels for addresses within clusters, but we can gain some insight that entities within the same cluster do have similar features. The similarity of entities in the same service is more obvious, and if using it as a constraint on the classification results of individual addresses, individual addresses that do not match the service performance can be corrected, thus improving the accuracy of the overall prediction. Note that we should still focus on the classification results of individual addresses, and the entity label correction only serves as a constraint. To implement the above ideas, the open and transparent nature of Bitcoin and the existing mature machine learning methods can be of good help, however, two efforts remain to improve the accuracy of the predictions as much as possible. (1) How to accurately describe the behavior patterns of addresses through the transaction history of individual addresses? In this paper, based on previous research, the historical transactions of Bitcoin addresses and the complex structure of the inter-address transaction graph are described, and both explicit and implicit features are summarized, fully exploit the features of addresses that contain specific semantics. (2) How to choose a machine learning model? In this paper, we use the current best ensemble learning models Random Forest and Gradient Boosting as the comparison models and use their outputs as the input for the consensus model. In contrast, unsupervised clustering uses spectral clustering, as it is suitable for dealing with sparse data with low dimensionality and a small number. The contribution of this paper can be summarized as follows:

1. In this paper, we summarize previous research and conduct comprehensive mining of features based on a single address. 110 features are extracted in this paper, including explicit features based on address transaction history and implicit features describing connections between addresses.
2. In this paper, we propose a consensus scheme for correcting address supervised classification results with entity unsupervised clustering results, which, to our knowledge, is the first scheme to combine multivariate model joint prediction with the Bitcoin multi-class service detection problem.
3. In this paper, the features mined are comprehensively analysed and the proposed method is compared and analysed with advanced machine learning methods to demonstrate the effectiveness of the solution.

3 Proposed Method

The address service identification approach proposed in this paper can be summarized in four steps: 1) retrieving and preprocessing historical transactions of addresses, 2) feature extraction, 3) training baseline models, and 4) consensus of results at the output level.

3.1 Transaction Retrieval and Pre-Processing

In this paper, we use the labeled address dataset [21] published in the article [16]. There are a total of 26,309 addresses and labels, of which the labels are divided into seven categories, namely Exchange, Faucet, Gambling, HYIP, Market, Mixer, and Mining Pool. The descriptions for the services are shown in Table 1. We first retrieve the full history of transactions for the address and pre-process the transaction information using sampling and exchange rate conversion. Sampling was used to retrieve historical transactions because we found that some addresses had even more than 400,000 transactions, which would be wasteful for analysing the behaviour of that address and would significantly increase

the time for feature extraction, which is not conducive to training a timely and usable model, so we randomly sampled no more than 500 transactions for each address. The conversion of the exchange rate is mainly due to the high volatility of the BTC price, as shown in Fig. 1, the extracted features related to the value will be unstable, thus having a negative impact on the classifier, so we use the API provided by coindesk [22] to import the exchange rate of USD and Bitcoin to ensure the stability of the value features.

Table 1: A list of the main services operated with Bitcoin

Service	Description
Exchange/wallet	Exchanging among fiat currencies and Bitcoin and manages users' Bitcoin.
Faucet	Offering free but small amount of Bitcoin in return for solving CAPTCHA, or clicking advertisements.
Gambling	Gambling games, e.g., dice and roulette.
HYIP	Investment program that promises high interest return, e.g., 1% per day.
Marketplace	Payment service, e.g., escrow, is offered in an online marketplace.
Mining pool	Cooperative mining team that shares computational power to find blocks. If one of the pool members finds a block, its minted Bitcoin is shared by them.
Mixer	Laundering a several Bitcoin transactions to avoid Bitcoin flow tracking.

3.2 Feature Extraction

This paper mines address features from two perspectives, which can be broadly divided into explicit features that summarize historical transactions and implicit features that describe relationships between addresses. Among them, the explicit feature extraction method retrieves all historical transactions of an address to get single features such as balance, lifetime, the total number of transactions, etc. In addition, we also summarize the historical transactions of an address, bring together data such as the amount of each transaction into a list, and obtain its statistical features such as maximum and minimum values, mean value, variance, standard deviation, etc. In this paper, 92 explicit features are extracted.

The implicit feature is then extracted based on [23], as shown in Fig. 5, where we make the following definition. The black circle represents the address A to be analyzed. Taking a set of transactions as an example, when A is used as the output of a transaction, all input addresses of that transaction are called ancestor addresses of that address, i.e., the address connected by the blue line in the figure. And so on, the addresses connected by yellow, green, and red are output sibling address, input sibling address, and successor address, respectively. The figure describes only two transactions in which the target address is involved, and we collect the neighbouring addresses of all its transactions and compute 12 quantity-related features by means of a merge set operation on the neighbouring addresses. The transaction patterns in Fig. 5 can then be summarised by means of Fig. 6, where the number of occurrences of six transaction patterns can be computed in a 2-motifs graph. The final total of 18 implied features is obtained. The descriptions of all extracted features are shown in Table 2. And finally, a small number of computationally convenient entity features used to normalise the results were extracted: *number of addresses within an entity*, *entity lifetime*, and *number of entity transactions*, respectively.

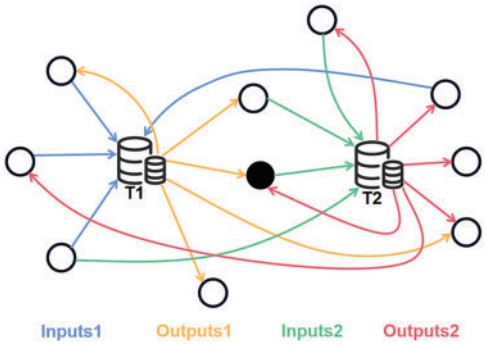

Figure 5: Diagram of the neighborhood address in the Bitcoin directed hypergraph

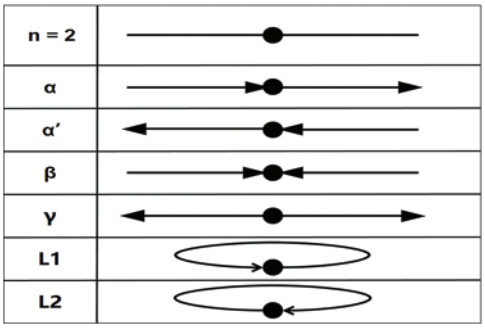

Figure 6: Trading patterns in the 2-motifs diagram

Table 2: The list of features extracted in this paper

Categories		Description
Explicit features	Unique features	The unique features of the address. The (lifetime, number) of (spent, received, all) transactions of the address, the address balance, the percentage of the total amount of the different indices, for a total of 27 features.
	Composite features	Statistical features of the address transaction sequence. Statistical features (maximum, minimum, mean, variance, standard deviation) of (dollar amount, byte sizes of transactions, inter-transaction intervals, block heights of the transactions, number of output/input transactions when used as an input/output address, ratio of transaction costs relative to the average transaction costs for the day) for (spend, receive, all) transactions at the address, respectively, for a total of $13 * 5 = 65$ features.
Implicit features	Quantitative features	A subgraph feature describing the neighbourhood relationship between addresses. The number of (unique, duplicate, all) addresses of (previous output, input, output, next input) addresses, i.e., (ancestor, input sibling, output sibling, descendant) addresses, when the address is involved in a transaction as an (input, output) address, respectively. For a total of 12 features.
	Pattern features	α, α', β, γ, $L1$, $L2$, respectively. For a total of 6 features.

3.3 Results Consensus Model

The inclusion of unsupervised clustering models in the classification combination can increase the diversity of the models, thus improving the accuracy and robustness of the predictions [19]. This paper considers the similarity of entity behaviors of the same services and proposes a scheme to correct the supervised classification results of address with the unsupervised clustering results of the entity. The method is a good remedy for the shortcomings of supervised or unsupervised learning only. For supervised learning, since Bitcoin addresses are generated without cost, some addresses

that are used only a few times do not match the features of their participating services and thus are inevitably misclassified, while entities are representational as mappings of real participants that are not easily generated or discarded. For unsupervised learning, clustering methods can only obtain the indexes of clusters, not the labels directly, and it is too absolute to simply group the addresses within a cluster into one category. However, the entities within the clusters obtained by unsupervised clustering have some correlation, which can be exploited for joint prediction in combination with supervised learning classifiers, and the entities are the ties that associate address objects of the same service. The ensemble problem can be viewed as an optimization problem on a bipartite graph, where the misclassified addresses in the entities are corrected by iterative propagation of probability estimates between neighboring entities. Achieving this optimization problem requires two things: First, the labels of the predictions should be consistent with the baseline of the supervised learning results as much as possible, which can be achieved by penalizing deviations from the predictions provided by supervised learning in an abstract bipartite graph optimization problem. Second, the unsupervised constraint needs to be satisfied to the maximum extent possible, and stable entity features replace their internally fluctuating address features for iterative propagation of probability estimates until they are stable and finally, a smooth prediction result is obtained.

In this paper, we use the following identifiers to denote the important parameters in the algorithm, the set of addresses to be predicted is $X = x_1, x_2, \ldots, x_n$, the addresses participate in c service categories. A total of m models are involved in joint prediction, where r models are supervised models that provide prediction data and the remaining $m - r$ unsupervised models provide clustering *ids* for the addresses. In a Bitcoin address multi-classification task, each model divides the data points into groups, so there will be a total of $v = m * c$ groups. The supervised learning model divides the addresses into $s = r * c$ groups with the same predictive labels, while the unsupervised model divides the addresses into $v - s$ groups with similar behavior. Note that the cluster *id* number z may not be the same as the category z because cluster *id* only distinguishes between different clusters to provide category constraints, and does not represent a specific service classification. In this paper, the $n \times m$ matrix is denoted by $B_{n\times m}$, and b_{ij} denotes the elements in row i and column j of the matrix. $\vec{b}_{i\cdot}$ and $\vec{b}_{\cdot j}$ denote the vectors of the $i - th$ row and $j - th$ column, respectively. If x_i is assigned to group g_j by one of the algorithms then $a_{ij} = 1$, otherwise 0. The matrix $A_{n\times v}$ formed by a_{ij} is called the affinity matrix.

Our goal is to compute the conditional probability u_{iz} that each address node x_i belongs to category z and the conditional probability q_{jz} that each group node g_j belongs to category z. The conditional probability matrix consisting of these two conditional probabilities is denoted as $U_{n\times c}$ and $Q_{v\times c}$, respectively. Since the first s groups are obtained from the supervised learning model and they have some initial class label estimates, we use $y_{jz} = 1$ to indicate that the group node g_j belongs to the class z and 0 otherwise, and the matrix consisting of y_{jz} is denoted as $Y_{v\times c}$. Finally, the number of categories assigned to each group is denoted as k_j and $k_j = \sum_{z=1}^{c} y_{jz}$. Table 3 summarizes the important notations.

Table 3: Description of symbols

Symbol	Definition
$1, ..., c$	Class indexes
$1, ..., n$	Object indexes
$1, ..., s$	Indexes of groups from supervised models

(Continued)

Table 3 (continued)

Symbol	Definition
$s+1, ..., v$	Indexes of groups from unsupervised models
$A_{n\times v} = [a_{ij}]$	a_{ij}—indicator of object i in group j
$U_{n\times c} = [u_{iz}]$	u_{iz}—probability of object i with respect to class z (i.e., $u_{iz} = \hat{P}(y = z \mid x_i)$)
$Q_{v\times c} = [q_{jz}]$	q_{jz}—probability of group j with respect to class z (i.e., $q_{jz} = \hat{P}(y = z \mid g_j)$)
$Y_{v\times c} = [y_{jz}]$	y_{jz}—indicator of group j predicted as class z

To reach a consensus among all models, an optimization problem is defined in a bipartite graph with an objective function that penalizes deviations from the base classifier predictions and differences in predicted class labels between nearby nodes. As Eq. (1) is shown, where $|\cdot|$ and $||\cdot||$ denote the L1 and L2 paradigms of the vector, respectively.

$$\min_{Q,U} \varphi(Q, U) = \min_{Q,U} \left(\sum_{i=1}^{n} \sum_{j=1}^{v} a_{ij} ||\vec{u}_{i\cdot} - \vec{q}_{j\cdot}||^2 + \alpha \sum_{j=1}^{v} k_j ||\vec{q}_{j\cdot} - \vec{y}_{j\cdot}||^2 \right) \tag{1}$$

$$\text{s.t. } \vec{u}_{i\cdot} \geq \overrightarrow{0}, |\vec{u}_{i\cdot}| = 1, i = 1 : n \qquad \vec{q}_{j\cdot} \geq \overrightarrow{0}, |\vec{q}_{j\cdot}| = 1, j = 1 : v$$

The optimisation method consists of two parts: The first part of the formula is to ensure that if an address object x_i is assigned to the group g_j by one of the classification algorithms, i.e., when $a_{ij} = 1$, their conditional probability estimates must be close to each other. That is, the group conditional probability of an address is similar to that of the entity to which it belongs, and this condition corrects the classification results of supervised learning and reduces the number of misclassified addresses. The second part is the constraint that the consensus class label estimate for the group g_j should not deviate significantly from its initial class label prediction y_j. This means that the address-corrected group conditional probabilities must not deviate too much from the initial predicted labels, so that only a small number of obviously misclassified addresses are modified, which ensures that the algorithm is dominated by the classification results. In this formula, α is the penalty for violating that constraint, The group node g_j participates in the constraint only if $j = 1, \ldots, s$, because it is generated by the classifier with $k_j = 1$, and the group node generated by unsupervised clustering does not participate in this constraint. Finally, $\vec{u}_{i\cdot}$ and $\vec{q}_{j\cdot}$ are probability vectors, so each component must be greater than or equal to 0 and sum to 1.

This optimisation problem can be solved using coordinate descent. The basic idea is to transform the optimisation problem of a multivariate function into an optimisation problem of multiple univariate functions, so that each variable is solved individually for optimality, and iterations are repeated until convergence to the optimal solution of the function. In Eq. (1), at the tth iteration, If the value of U is fixed, the objective function is a summation of v quadratic components with respect to $\vec{q}_{j\cdot}$. It is strictly convex and $\nabla_{\vec{q}_{j\cdot}} \varphi(Q, U^{(t-1)}) = 0$ gives the unique global minimum of the cost function with respect to $\vec{q}_{j\cdot}$:

$$\vec{q}_{j\cdot}^{(t)} = \frac{\sum_{i=1}^{n} a_{ij} \vec{u}_{i\cdot}^{(t-1)} + \alpha k_j \vec{y}_{j\cdot}}{\sum_{i=1}^{n} a_{ij} + \alpha k_j}. \tag{2}$$

Similarly, fixing Q, the unique global minimum with respect to $\vec{u}_{i\cdot}$ is also obtained:

$$\vec{u}_{i\cdot}^{(t)} = \frac{\sum_{j=1}^{v} a_{ij}\vec{q}_{j\cdot}^{(t)}}{\sum_{j=1}^{v} a_{ij}}. \tag{3}$$

Algorithm 1: BGCM algorithm

Input: group-object affinity matrix $\boldsymbol{A}$, initial labeling matrix $\boldsymbol{Y}$; parameters α and ε
Output: consensus matrix $\boldsymbol{U}$

```
Initialize U^0, U^1 randomly
Set step number t ← 1
while ‖U^t − U^{t−1}‖ > ϵ do
Q^t = (D_v + αK_v)^{−1} (A^T U^{t−1} + αK_v Y)
U^t = D_n^{−1} A Q^t
end while
return U^t
```

The updated formula for the matrix is shown in Algorithm 1. The algorithm demonstrates a correction to the original probability estimation matrix (containing supervised learning results). The block coordinate descent is used to solve this optimisation problem. The basic idea is to transform the optimisation problem of a multivariate function into an optimisation problem of multiple univariate functions so that each variable is solved individually for optimality, and iterations are repeated until convergence to the optimal solution of the function.

In this algorithm, $D_v = diag\left\{\left(\sum_{i=1}^{n} a_{ij}\right)\right\}_{v\times v}$ and $D_n = diag\left\{\left(\sum_{j=1}^{n} a_{ij}\right)\right\}_{n\times n}$ act as the normalization factor, which maps the matrix after generating a change to a standard range. $K_v = diag\left\{\left(\sum_{z=1}^{c} y_{jz}\right)\right\}_{v\times v}$ indicates the existence of constraints on the group nodes, and parameter α can change the strength of this constraint. And t denotes the number of iterations. In the tth iteration, the probability estimation matrix of a group node (i.e., Q) combines the information of its neighbouring object nodes and its initial value Y. And when updating the consensus result U, it propagates its updated probability estimate back to its neighbouring object nodes. The results converge when $\left\|U^t - U^{t-1}\right\| \leq \varepsilon$, at which point the loop ends. And $\left(Q^{(t)}, U^{(t)}\right)$ both converge to the stationary point of the optimization problem. Reference [24] showed a simple example and intermediate results for matrix transformations.

4 Experiment and Evaluation

In this paper, we use Bitcoin-core [25] to obtain Bitcoin block information and use the Blocksci library [26] to process and retrieve historical Bitcoin transactions. The experimental code is written in python 3.7 and mainly uses the blocksci, pandas, numpy, and scikit-learn libraries. In this section, we evaluate the feature contribution, model accuracy, resulting confusion matrix and distribution of feature values across the service, respectively.

4.1 Comparison of Feature Extraction Methods

In this section, we experiment on the accuracy of the model under the use of two types of feature extraction methods, we divide the tests into accuracy comparisons using only explicit features, using only implicit features, and using all features. Random Forest was chosen as the machine learning algorithm, and the number of estimators was set to 100. Due to the imbalance in the number of addresses in the different categories in the dataset, we sampled the same number of addresses from

each category in each of the following classification experiments, i.e., 100 addresses were extracted from each category of data to form the dataset, so that the size of the dataset at each round of evaluation was 700. Each round of experiments uses the triple cross validation method, i.e., 2/3 of the data is used for training and 1/3 of the data is used for testing, and each round will produce three results. The experimental results are highly correlated with the division of the dataset, and the larger the share of the test set, the higher the accuracy. However, if the proportion of the training set is too large, it will lose its reference value, so this experiment uses the triple cross validation method. The rest of the experiments were carried out as described above.

To better show the difference, we repeated the accuracy assessment 100 times and plotted the accuracy as a line graph and the final result is shown in Fig. 7. Since the sample set of addresses is randomly sampled in each round, the accuracy rate fluctuates, and the average accuracy results are 0.754, 0.73, and 0.691 in that order. The high accuracy rate can be obtained with a small number of features using only implicit features, which indicates that the inter-address transaction relationship is an important factor. Fig. 8 shows the top 10 contributed features. As can be seen from this figure, the address *lifetime* is the most contributed feature and implicit features also occupy an important place.

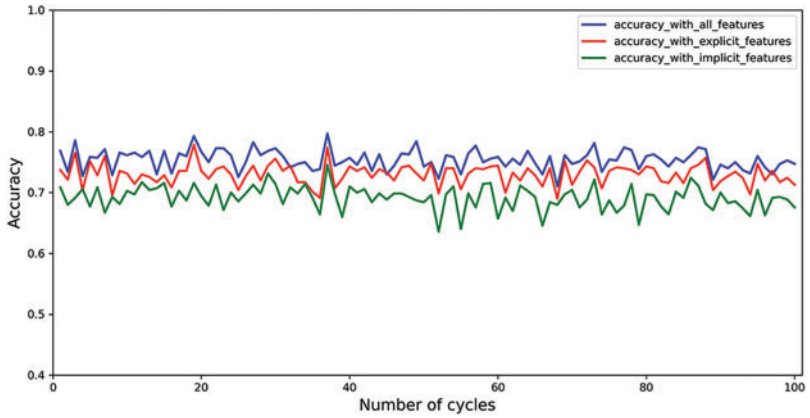

Figure 7: Comparison of prediction accuracy under different categories of features

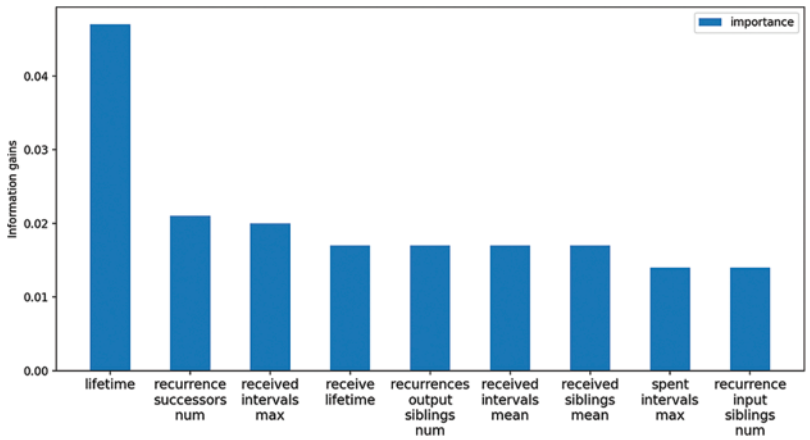

Figure 8: Feature importance

4.2 Model Accuracy Evaluation

In this experiment, BGCM is compared with the two best result algorithms, random forest and gradient boosting, to test the accuracy improvement. The number of estimators in the compared algorithms both set to 100, and the rest of the parameters are set to default values. These two results are used as inputs for the first two supervised learning results in the BGCM algorithm, and the rest of the unsupervised clustering inputs are generated by spectral clustering. The alpha parameter in the BGCM algorithm is set to 3, and the number of iterations is 30. Table 4 shows the accuracy, recall, and F1-score under different methods. Our method can obtain better classification results and can successfully correct the few addresses that are misclassified because they do not match the service features, thus improving the overall accuracy. In addition, our algorithm is somewhat pervasive and can be applied to other anonymous transaction situations similar to Bitcoin.

Table 4: Performance comparison of different algorithms

Methods	Precision	Recall	F1-score
Random forest	0.758	0.753	0.753
Gradient boosting	0.767	0.761	0.760
BGCM	**0.774**	**0.773**	**0.774**

4.3 Confusion Matrix

Table 5 shows only the confusion matrix of classification results based on the BGCM algorithm, where each row of the table represents the actual service, while each column represents the predicted service.

Table 5: Confusion matrix of the BGCM scheme

	Exchange	Faucet	Gambling	HYIP	Market	Mixer	Mining pool
Exchange	**0.63**	0.04	0.12	0.05	0.07	0.01	0.07
Faucet	0.04	**0.86**	0.06	0.06	0.01	0.0	0.01
Gambling	0.11	0.03	**0.58**	0.07	0.04	0.0	0.03
HYIP	0.04	0.05	0.07	**0.74**	0.01	0.01	0.06
Market	0.11	0.02	0.09	0.02	**0.84**	0.02	0.02
Mixer	0.02	0.0	0.03	0.01	0.02	**0.96**	0.0
Mining pool	0.05	0.0	0.05	0.05	0.01	0.01	**0.81**

It is easy to see that exchanges and gambling have the lowest detection accuracy, which may be due to the high number of addresses in the dataset for these two services, resulting in a wide distribution of their features and uneven sampling. Furthermore, there are many similarities between the addresses in these two categories, with 12% of exchange addresses being misclassified as gambling and 11% of gambling addresses being misclassified as exchanges, an acceptable result as both services transact with a large number of users and with variable transaction values. To further distinguish between exchanges and gambling services, more detailed features would need to be extracted. Nevertheless, our method still shows some improvement in classification effectiveness.

4.4 Distribution of Feature Values Across Services

To more visually see how these features differ by service, we selected the top six contributed features to plot their distribution by service using a box plot. The box plot is a good way to see the distribution of features by service. The top and bottom of the box and the horizontal lines inside the box represent the interquartile range. Two lines are drawn vertically from the top and bottom of the box, with the edges of the lines and black dots indicating unusual outliers. Thus, if the box is 'squashed' or if the box is short in length, it means that the feature values of the service are relatively specific, while if the box is large, it means that the feature values of the service are widely distributed and there is no clear pattern. The box plot reflects the distribution of features in a single dimension.

Fig. 9 shows a box plot of the most contributed features by service, with the red line representing the 50th percentile, the green triangle representing the mean, and the black origin representing the outliers. Fig. 9a shows the lifetime box plot distribution of addresses by different services. It is obvious that exchanges, gambling, marketplaces and faucets have longer address lifetime, especially exchanges, as exchange addresses provide trading services for a long time. Conversely, HYIP, mixer and mining pool addresses have a shorter lifetime. In the case of HYIP addresses, which are inherently risky and illegal to trade, most addresses within the IQR follow the use-and-discard principle of using one-time addresses for scams to reduce the risk of being traced, and it is evident that entities offering such services are very cautious. Although HYIP addresses mostly have a short lifetime, a certain amount of time exists as entities need time to crowdfund. In contrast, a coin mixer address is a true one-time use address, which is related to the nature of the coin mixing service. To add a greater degree of confusion, coin mixer services typically use many temporary addresses that are only used for a single transaction and which are not directly linked to the user's real identity or other transaction addresses. By using one-time addresses, coin mixers break the correlation between the original address and the final destination address. This makes transactions more difficult to trace and increases the privacy and anonymity of the user's transactions. The distribution of address lifetime for the "mining pool service" is rather peculiar, with the average of the mining pool address lifecycle being at the top of the box, i.e., most addresses have a short lifecycle, but a few "old miners" with a long lifecycle pull up the average. Our guess is that the pool model is now the mainstay of mining due to the increased difficulty of mining. The pools recruit miners on a temporary basis and reward them based on the power they contribute, so the majority of miners are now temporary miners with a short lifetime. But most of the old miners are still fighting on.

Fig. 9b reflects the distribution of the number of repeat successors across the different services. The meaning of this feature is that when the address is being used as a sending address, the number of times the address is repeated in the received address for all transactions. An interesting phenomenon emerges from the graph, where there is a large number of repeated successors to the faucet address. We suspect that in order to receive this modest reward, the participants actively participate and accumulate more. In contrast, coin mixers hardly ever have repeated transaction addresses for the purpose of user privacy protection. Differences in features between services can distinguish well between the services offered by addresses, and these features are often not directly related to the value of money, which requires mining deeper features of the relationship between addresses. This paper provides a clearer picture of the nature of each service by analysing the distribution of features by service.

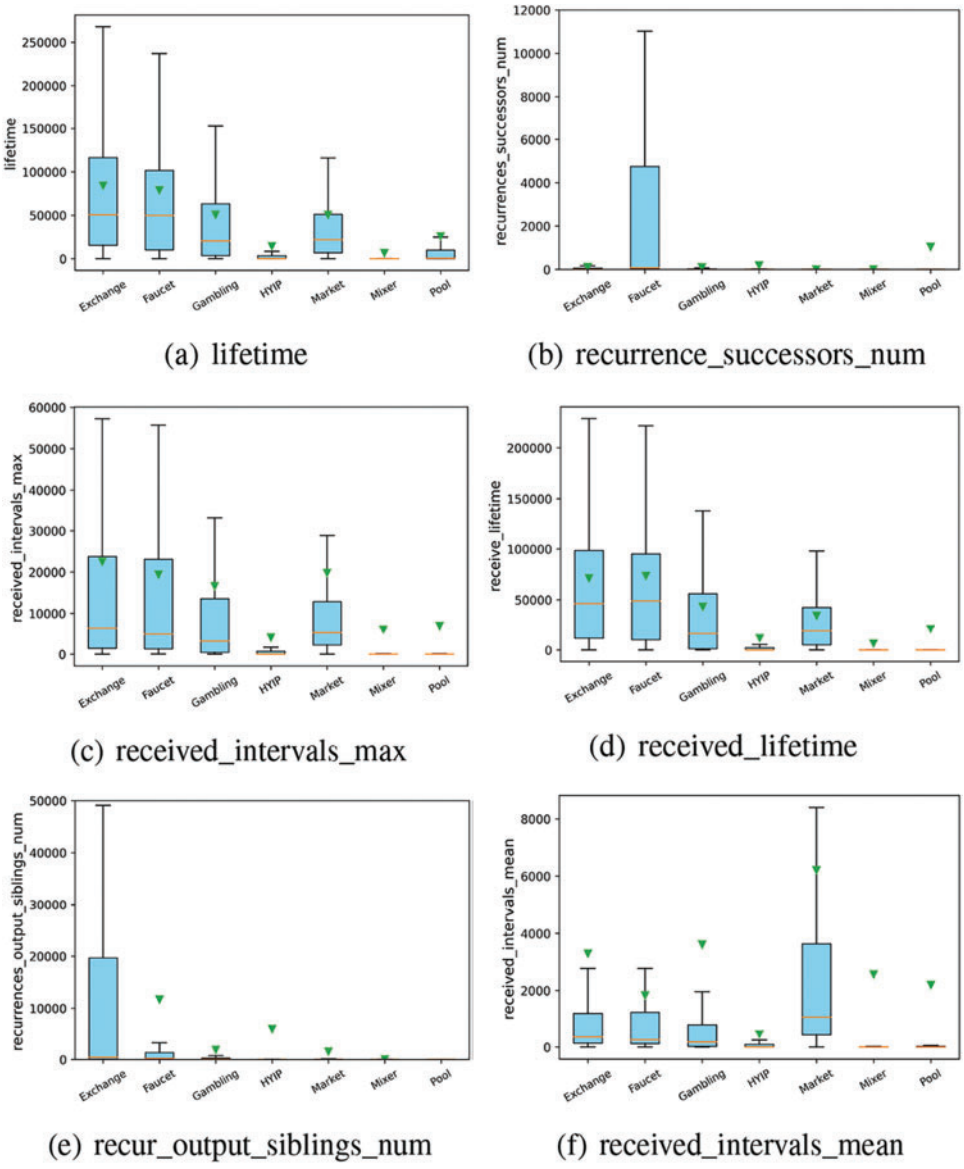

Figure 9: Boxplots of contributed features by services

5 Conclusion

In this paper, we propose a new approach to improve the prediction accuracy of Bitcoin address service classes by combining the features of addresses and the features of the entities they are located in to identify address services. Identifying address involved services has many benefits, such as dangerous transaction prevention and statistical analysis of Bitcoin services. Specifically, we show how a multi-model maximization consensus mechanism can be combined with Bitcoin addresses and entities to obtain better classification performance. With the tri-fold crossvalidation method, we can obtain a global average accuracy of 77.4%. In addition, up to 110 features with specific semantics are extracted, and the experimental results show that, in addition to addressing lifetime and transaction interval, the implicit features describing the transaction relationship between addresses are better for distinguishing

different services. We select the features with the highest contribution and interpret and analyse the experimental results by combining their semantic information and their distribution by services. We believe that analysing the features of each Bitcoin service can help us better understand the trends of Bitcoin changes. In future work, we would like to focus our work on studying the detection of suspicious addresses using unsupervised learning algorithms, because in real life, most Bitcoin addresses or transactions are made by normal users and it is difficult to obtain labels, using supervised learning algorithms tends to generate a large number of false positive addresses, so our idea is to use outlier detection algorithms to filter out a large number of normal nodes, and further perform outlier nodes for fraud category detection to minimize the number of misclassified users. Nevertheless, in this paper, we were able to classify the seven classes of addresses that have been tagged for services with high accuracy, and we believe that this research can help investigate crimes and assist law enforcement agencies in detecting illegal activities in the Bitcoin network.

Acknowledgement: The authors are thankful to the previous contributors of machine learning technology that used in this research.

Funding Statement: This work is sponsored by the National Natural Science Foundation of China Nos. 62172353, 62302114 and U20B2046. Future Network Scientific Research Fund Project No. FNSRFP-2021-YB-48. Innovation Fund Program of the Engineering Research Center for Integration and Application of Digital Learning Technology of Ministry of Education No. 1221045.

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Lejun Zhang, Junjie Zhang, Ran Guo; data collection: Kentaroh Toyoda; analysis and interpretation of results: Junjie Zhang, Yuan Liu, Zhihong Tian; draft manuscript preparation: Lejun Zhang, Junjie Zhang, Jing Qiu. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The datasets used and analysed during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review,* 21260.
2. Tasca, P., Hayes, A., Liu, S. (2018). The evolution of the bitcoin economy: Extracting and analyzing the network of payment relationships. *The Journal of Risk Finance, 19(2),* 94–126.
3. Zhang, R., Xue, R., Liu, L. (2019). Security and privacy on blockchain. *ACM Computing Surveys, 52(3),* 1–34.
4. Xu, J. J. (2016). Are blockchains immune to all malicious attacks? *Financial Innovation, 2(1),* 1–9.
5. Blockchaininfo. https://www.blockchain.com/explorer (accessed on 23/02/2023)
6. Möser, M., Böhme, R., Breuker, D. (2014). Towards risk scoring of bitcoin transactions. *International Conference on Financial Cryptography and Data Security*, Barbados, Springer.
7. Tian, Z., Li, M., Qiu, M., Sun, Y., Su, S. (2019). Block-DEF: A secure digital evidence framework using blockchain. *Information Sciences, 491(3),* 151–165.
8. Wang, W., Hu, N., Liu, X. (2019). Blockzone: A blockchain-based DNS storage and retrieval scheme. *International Conference on Artificial Intelligence and Security*, New York, USA, Springer.

9. Su, S., Tian, Z., Li, S., Deng, J., Yin, L. et al. (2021). IoT root union: A decentralized name resolving system for IoT based on blockchain. *Information Processing & Management, 58(3),* 102553.
10. Androulaki, E., Karame, G. O., Roeschlin, M., Scherer, T., Capkun, S. (2013). Evaluating user privacy in bitcoin. *International Conference on Financial Cryptography and Data Security*, Okinawa, Japan, Springer.
11. Bonneau, J., Narayanan, A., Miller, A., Clark, J., Kroll, J. A. et al. (2014). Mixcoin: Anonymity for bitcoin with accountable mixes. *International Conference on Financial Cryptography and Data Security*, Barbados, Springer.
12. Wu, L., Hu, Y., Zhou, Y., Wang, H., Luo, X. et al. (2021). Towards understanding and demystifying bitcoin mixing services. *Proceedings of the Web Conference 2021*, Ljubljana, Slovenia.
13. Meiklejohn, S., Pomarole, M., Jordan, G., Levchenko, K., McCoy, D. et al. (2013). A fistful of bitcoins: Characterizing payments among men with no names. *Proceedings of the 2013 Conference on Internet Measurement Conference*, Barcelona, Spain, IEEE.
14. Akcora, C. G., Li, Y., Gel, Y. R., Kantarcioglu, M. (2019). Bitcoinheist: Topological data analysis for ransomware detection on the bitcoin blockchain. arXiv preprint arXiv:1906.07852.
15. Jourdan, M., Blandin, S., Wynter, L., Deshpande, P. (2018). Characterizing entities in the Bitcoin blockchain. *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, Singapore, IEEE.
16. Toyoda, K., Ohtsuki, T., Mathiopoulos, P. T. (2018). Multi-class bitcoin-enabled service identification based on transaction history summarization. *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, Bhimtal, India, IEEE.
17. Zola, F., Eguimendia, M., Bruse, J. L., Urrutia, R. O. (2019). Cascading machine learning to attack bitcoin anonymity. *2019 IEEE International Conference on Blockchain (Blockchain)*, Atlanta, USA, IEEE.
18. Kanemura, K., Toyoda, K., Ohtsuki, T. (2019). Identification of darknet markets' bitcoin addresses by voting per-address classification results. *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, Seoul, South Korea, IEEE.
19. Gao, J., Liang, F., Fan, W., Sun, Y., Han, J. (2009). Graph-based consensus maximization among multiple supervised and unsupervised models. *Advances in Neural Information Processing Systems, 22,* 585–593.
20. Yang, L., Dong, X., Xing, S., Zheng, J., Gu, X. et al. (2019). An abnormal transaction detection mechanim on bitcoin. *2019 International Conference on Networking and Network Applications (NaNA)*, Daegu, South Korea, IEEE.
21. dataset_IEEEBlockchain2018. https://goo.gl/sQJKdx (accessed on 23/02/2023)
22. Coindesk. https://www.coindesk.com/price/ (accessed on 23/02/2023)
23. Ranshous, S., Joslyn, C. A., Kreyling, S., Nowak, K., Samatova, N. F. et al. (2017). Exchange pattern mining in the bitcoin transaction directed hypergraph. *International Conference on Financial Cryptography and Data Security*, Netherlands, Springer.
24. Gao, J., Liang, F., Fan, W., Sun, Y., Han, J. (2011). A graph-based consensus maximization approach for combining multiple supervised and unsupervised models. *IEEE Transactions on Knowledge and Data Engineering, 25(1),* 15–28.
25. Bitcoin Core. https://bitcoin.org/en/bitcoin-core/ (accessed on 23/02/2023)
26. Blocksci. https://citp.github.io/BlockSci/ (accessed on 23/02/2023)