A New Bitcoin Address Association Method Using a Two-Level Learner Model

Tengyu Liu[1,2], Jingguo Ge[1,2](✉), Yulei Wu[3], Bowei Dai[4], Liangxiong Li[1,2], Zhongjiang Yao[1,2], Jifei Wen[1,2], and Hongbin Shi[1,2]

[1] Institute of Information Engineering, Chinese Academy of Sciences,
Beijing 100093, China
gejingguo@iie.ac.cn

[2] School of Cyber Security, University of Chinese Academy of Sciences,
Beijing 100049, China

[3] College of Engineering, Mathematics and Physical Sciences, University of Exeter,
Exeter EX4 4QF, UK

[4] Institute of Microelectronics of the Chinese Academy of Sciences,
Beijing 100029, China

Abstract. Users in the Bitcoin system adopt a pseudonym-Bitcoin address as the transaction account, making Bitcoin address correlation analysis a challenging task. Under this circumstance, this paper provides a new Bitcoin address association scheme which makes address tracing possible in Bitcoin systems. The proposed scheme can be used to warn relevant institutions to study more secure encryption algorithms to protect users' privacy. Specifically, the important features of a Bitcoin address are extracted. After that, to reduce the computational complexity, we transform the clustering problem of addresses into a binary classification problem in which we integrate the features of two Bitcoin addresses. A novel two-level learner model is then built to analyze if the two Bitcoin addresses are belonging to the same user. Finally we cluster the addresses belonging to the same user accordingly. Extensive experimental results show that the proposed method outperforms the other address association schemes, which use deep learning models or heuristics, and can achieve an increase by 6%–20% in precision and by 10% improvement in recall.

Keywords: Bitcoin · Blockchain · Two-level learner · Bitcoin security

1 Introduction

Bitcoin is the first successful implementation of a digital currency that enables instant payments to anyone, anywhere in the world. The Bitcoin system is designed based on the idea of using a decentralized peer-to-peer network, rather than relying on central authorities. Blockchain is the basis of the Bitcoin system,

S. Wen et al. (Eds.): ICA3PP 2019, LNCS 11945, pp. 349–364, 2020.
https://doi.org/10.1007/978-3-030-38961-1_31

which provides a distributed infrastructure and an anonymous way of trading to guarantee the security and freedom of Bitcoin. There is no central server in the network. It uses distributed nodes to generate, update, and store data. Furthermore, it employs cryptography for data transmission and provides a secure and credible environment for Bitcoin transactions.

In details, when users use Bitcoin to trade with others, these transactions will be verified by the nodes in the blockchain network and then be recorded on the blockchain. To enhance privacy and security, users adopt pseudonyms-Bitcoin addresses [1] as their transaction accounts to send and receive bitcoins. The user's real identity will not be publicly bound to the transaction account. In principle, each user can have hundreds of different Bitcoin addresses.

This pseudo-anonymity nature has been improperly used by illegal activities, such as money laundering [2,3] and ransomware [4], to circumvent supervision. Recently, many researchers focus on Bitcoin de-anonymization to address this issue. De-anonymization makes Bitcoin address tracing possible and helps the regulatory system implement network security management. In addition, it can also warn relevant institutions to study more secure encryption algorithms to protect users' privacy.

Many heuristics and deep learning methods have been investigated to do Bitcoin de-anonymization. However, the performance of these methods were still not satisfactory. To make the Bitcoin de-anonymization more accurate and simple-to-implement, in this paper we resort to the important information of Bitcoin addresses based on the historical transactions and manage to associate the addresses. However, there exist many challenges because of the privacy issues: (1) The real identity of the owner of Bitcoin addresses involved in the transaction is unknown because of the anonymity. Thus, few features of Bitcoin addresses can be obtained to analyze. (2) Users may generate fresh Bitcoin addresses for each transaction. (3) It is difficult to apply the clustering analysis of Bitcoin addresses to associate them, because of the large number of users and massive categories to be classified.

To this end, we propose a new Bitcoin de-anonymization method, Bitcoin address association, in which we combine Bitcoin addresses in pairs and then use a two-level learner to determine whether two Bitcoin addresses belong to the same user. The main contributions of this paper can be summarized as follows:

- The large number of clustering categories increases the complexity of clustering algorithms. To tackle this problem, we transform the clustering problem into a binary classification problem to categorize the address pairs.
- In order to gain better classification results, we propose a new model, which is a two-level learner to classify Bitcoin address pairs. XGBoost, LightGBM and Gradient boosting decision tree (GBDT) are used in the first-level learner because they can handle all kinds of features well (see Sect. 3.3 for details). Deep neural network (DNN) is then employed in the second-level learner due to its excellent performance on classification (details in Sect. 3.3).

- Extensive experimental results are conducted to compare the performance of our proposed model with other existing methods. The results demonstrate that our model outperforms the existing methods, with almost 6%–10% improvement in precision and almost 20% improvement in recall.
- We further analyze the effect of Bitcoin address combination orders on the performance of the model. The results show that the combination order of two groups of addresses has little effect on the model's performance. It is therefore not necessary to consider the combination order of the two addresses when using our model for address association.

The rest of this paper is organized as follows. Section 2 gives a minimalistic introduction about related work on Bitcoin address de-anonymization. Section 3 presents the main method of how to associate two Bitcoin addresses. Section 4 shows the experimental results and carries out the performance analysis. Finally, we draw conclusions in Sect. 5.

2 Related Work

With the continuous development and maturity of the Bitcoin system, the number of Bitcoin users is increasing. Researchers began to study the de-anonymization of Bitcoin addresses mainly through three ways. One is clustering analysis on the basis of heuristics, another is analyzing the structure and characteristics of the underlying distributed system, and the last one is based on deep learning methods.

Clustering Analysis. In [5], the authors proposed two heuristic methods. One is "multi-input" heuristic, which is the most effective and simplest method. This method assumes that all input addresses participating in the same transaction belong to the same user. The second is "shadow" address. Assuming that a user seldom trades with two different users, the Bitcoin address storing the change resulted from a transaction also belongs to the input user of the transaction. The authors can easily map any Bitcoin addresses and users through the above two heuristic analysis of large transactions in the Bitcoin trading network. Fleder et al. [6] crawled the Bitcoin addresses and transaction information fragments from web forums. They then tracked users' whereabouts and conducted clustering analysis, so as to associate transactions and users, and achieve the purpose of de-anonymity. The authors in [7] introduced an efficient automatic cluster approach which used off-chain information as votes for address separation, and then considered it together with blockchain information obtained during the clustering model construction step. In [8], Nick combined the above two heuristics with other heuristics, and the mean recall for address mapping is approximately 0.709.

Using Distributed Networks. Several researches are performed to track the source of Bitcoin addresses by observing and analyzing the structure and characteristics of Bitcoin's underlying distributed networks. In [9], the authors used the open trading history of Bitcoin to establish the transaction network between addresses and between users. They combined these structures with external information and techniques such as context discovery and flow analysis to investigate an alleged theft of Bitcoin. The studies in [10] and [11] tracked user's identity information with distributed network characteristics and demonstrated the power of data mining technology in the de-anonymization of Bitcoin addresses. Sybil Attack in [12] and Fake Node Attack in [13] used the users' IP addresses to perform de-anonymization analysis. Mastan [14] proposed a new approach to link the sessions of unreachable Bitcoin nodes, based on the organization of the block-requests made by the nodes in a Bitcoin session graph with a precision of 0.90 and a recall of 0.71.

Deep Learning Methods. In [15], Shao, Li and Chen designed a pipeline for Bitcoin address featuring that converts raw address features into a primary address vector. Then they employed a deep learning system and k-NearestNeighbor algorithm that realize Bitcoin addresses clustering progressively whose precision achieves 0.766 and recall is 0.836.

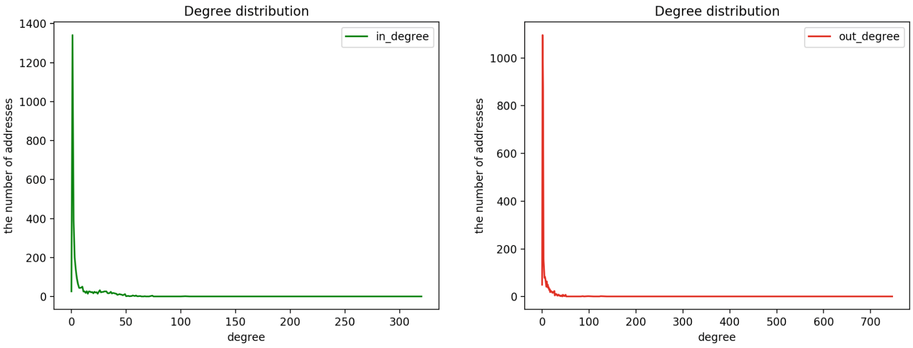
Although many de-anonymization technologies for tracing illegal activities and strengthening supervision have been proposed, their performance is still not satisfactory and the risks still exist.

3 The Proposed Method

The main workflow of this article is as follows. We extract each Bitcoin address' base features (in Sect. 3.1) and pre-process the samples of Bitcoin addresses (in Sect. 3.2). We then use the GBDT, XGBoost and LightGBM mechanisms and the Function L to learn the new features, which are then fed into a DNN network (in Sect. 3.3) for further analysis. Finally, we cluster the addresses based on the relationship between them.

3.1 Basic Feature Analysis and Extraction

We analogize the transactions between Bitcoin addresses to those between users on e-commerce platforms. The user characteristics obtained from e-commerce platforms are mainly based on the users' behaviors, including transaction time, transaction amount, transaction times and other characteristics [16]. The transactions between Bitcoin addresses can be treated as a network, which we call a Bitcoin transaction network. Bitcoin addresses are the network nodes, and the transaction amount flows are the network links. When the address acts as a money receiver, the number of links pointing to it is called in-degree. Out-degree is the contrary. We analyze the in-degree and out-degree distributions of Bitcoin addresses in this network as shown in Fig. 1, which illustrates the number of Bitcoin addresses of in-degree and out-degree. From Fig. 1, we can find that there are a large number of small degree nodes in the network, and also a considerable number of high degree nodes (Hub nodes). The degree can



(a) in-degree

(b) out-degree

Fig. 1. The degree distribution of a Bitcoin transaction network: (a) represents the in-degree distribution of Bitcoin addresses which act as a money receiver, and (b) represents the out-degree distribution of Bitcoin addresses which act as a money sender.

therefore be considered as a key feature to distinguish the addresses of different users.

To enhance the security, one user may create multiple Bitcoin addresses and use different addresses for transactions. Many addresses are used only for a short time, such as "shadow address" in [5]. Different users may use Bitcoin addresses differently, so we consider the characteristics related to the lifetime of a Bitcoin address. 18 features of a Bitcoin address are exacted according to [17], which are shown in Table 1.

3.2 Clustering to Binary-Classification Transformation

A straightforward way to identify which addresses belong to the same user is addresses clustering analysis [18]. Each user represents a category. Bitcoin addresses belonging to the same user are classified into the same category. When the number of users go up to a certain scale, it increases the difficulty and complexity of clustering algorithms and reduces the accuracy of clustering results.

In view of the problem that there are a huge number of categories to be classified in the address association problem, we combine two Bitcoin addresses into a new sample, instead of directly using a single address as a classification sample according to the suggestion from social network account association [19]. The purpose of our approach is to perform the classification in Bitcoin address pairs. In this way, we transform a clustering problem into a binary classification problem to categorize the address pairs. The experiments in Sect. 4 illustrate this new method can alleviate the difficulty in the clustering problem. The combination method is as follows.

Table 1. The features and its interpretation of each bitcoin address

Feature	Interpretation
Lifetime	The lifetime of each address
activity_days	The number of days that the address has participated in at least one transaction
max_trans_per_day	The maximum number of daily transactions of each address
total_received	The values of each address received
total_sent	The values sent out by each address
avg_val	The average of the values transformed from/to each address
std_val	The standard deviation of the values transformed from/to each address
in_gini	The Gini coefficient of the values transformed to the address
out_gini	The Gini coefficient of the values transformed from the address
in_trans_num	The number of transactions which the address acts as the input address
out_trans_num	The number of transactions which the address acts as the output address
ratio_btw_in_out	The ratio between in_trans_num and out_trans_num
in_digree	The number of addresses which have transferred money to the address
out_digree	The number of addresses which have received money from the address
max_time_diff	The maximum delay between the time when the address has received money and the time it has sent out to some others
min_time_diff	The minimum delay between the time when the address has received money and the time it has sent out to some others
avg_time_diff	The average delay between the time when the address has received money and the time it has sent out to some others
balance_two_days	The maximum difference between the balance of the address in two consecutive days

We use $\boldsymbol{a_p} = (x_1, x_2, \ldots, x_i, \ldots, x_n)$ representing the address p's eigenvector, and use $\boldsymbol{a_q} = (y_1, y_2, \ldots, y_i, \ldots, y_n)$ representing the address q's eigenvector. x_i and y_i denote the i-dimensional feature of $\boldsymbol{a_p}$ and $\boldsymbol{a_q}$, respectively. Given the addresses p and q, we can construct the address pair $(\boldsymbol{a_p}, \boldsymbol{a_q}) = (x_1, x_2, \ldots, x_i, \ldots, x_n, y_1, y_2, \ldots, y_i, \ldots, y_n)$. Then the relationship between addresses can be determined by a binary classification problem.

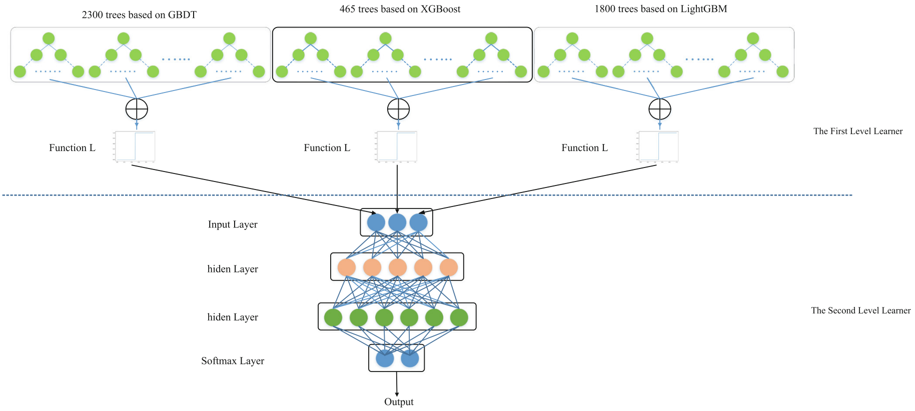


Fig. 2. The Two-level Learner Architecture: GBDT, XGBoost, LightGBM mechanism and Function L are used to perform the classification work in the first-level learner. Then the results are fed into the second-level learner which includes a DNN model to output the final results.

3.3 Model Stacking Architecture: A Two-Level Learner

Deep learning models [20] have been successfully applied in many fields and have performed well in classification problems. The structure of these models can be adjusted according to the experimental results.

Tree boosting is an effective and widely used machine learning method, due to its strong generalization ability and the capability of well-handling all kinds of features. Gradient boosting decision tree (GBDT) [21] is a typical tree boosting model, which can improve the accuracy of the final classifier by reducing the deviation in the training process. The experimental results in Sect. 4.3 show that it performs better than DNN [20] in classification problems. XGBoost [22] is a scalable end-to-end tree boosting system, which is widely used by data scientists to achieve state-of-the-art results on many machine learning challenges. Especially, it can automatically use the multi-threads of a CPU to parallelize the computation tasks and improve the accuracy on the basis of GBDT. LightGBM [23] is an efficient tree boosting model with higher computational speed and less memory consumption than GBDT and XGBoost. The experiments in Sect. 4.3 show that the three models perform well in the classification problems of Bitcoin addresses.

Stacking is a model ensembling technique used to combine information from multiple predictive models to generate a new model. Often, the stacked model (also called two-level learner) outperforms each of the individual models due to its smoothing nature, and the ability to highlight each base model on its best performance cases [24].

The model stacking is therefore employed in this paper to construct a two-level learner. The first-level learner is trained by the initial dataset, and the output is regarded as the input features of the second-level learner; the labels of the initial samples are still regarded as the labels of the second-level learner. The above three tree boosting models are considered as the first-level learner to get preliminary results. A DNN model is adopted in the second-level learner; its input features obey binomial distribution, and each input feature has the dimensions with the same order of magnitude. The experiments in Sect. 4.3 demonstrate that the use of DNN as the second-level learner outperforms that only using DNN as a classifier. The model architecture is shown in Fig. 2.

The First-Level Learner. As Fig. 2 shows, the first-level learner employs three tree models using the principle of GBDT, XGBoost, and LightGBM. At first, 2300 trees are built with the maximum depth of 8, which are based on GBDT. Then 465 trees are constructed using XGBoost mechanism; the maximum depth of the tree model is 15. Lastly we use the LightGBM mechanism to generate 1800 trees, and each tree has at most 34 leaves. These hyper-parameters are obtained experimentally. Because the outputs of the three models are decimal in $[0, 1]$, we employ a **function L:** $y = \begin{cases} 0 & 0 \leq x \leq 0.5 \\ 1 & 0.5 < x \leq 1 \end{cases}$ to transform the output of each model to 0 or 1, where x stands for the outputs of the three models, and y stands for the final results of the first-level learner. We then get the three corresponding outputs marked as (O_1, O_2, O_3).

The Second-Level Learner. The second-level learner is based on DNN, which consists of three fully connected layers. As for the design of the first layer, we adopt five units and the input is the output, (O_1, O_2, O_3), of the first-level learner. ReLU [25] is employed as the activation function. The output is sent to the second layer, which has six units and also adopts ReLU as the activation function. Then, we take the output of the second layer to the last layer with two units. At this layer, we employ Softmax [26] as the activation function. The number of units in each layer are confirmed through experiments. The details of DNN is summarized in Table 2.

Table 2. The details of deep neural network part

Layer	Input	Output	Activation function
First	3*1	5*1	ReLU
Second	5*1	6*1	ReLU
Third	6*1	2*1	Softmax

4 Experimental Results and Analysis

4.1 Dataset

The dataset in [27,28] which contains the transaction history in Mt.Gox is used in this study. This data set records the user's access to Bitcoin on the platform.

The user uses WalletID as the identity on the Mt.Gox platform. When the user needs to send or receive bitcoin, he only needs to initiate a request to Mt.Gox, and Mt.Gox helps the user complete the transaction and will record (WalletID, Entry, Date, Amount) on the platform. Each row in the dataset corresponds to a transaction of a user, and the transaction is simultaneously recorded on the blockchain with the bitcoin addresses of the user or Mt.Gox as input or output. According to the transaction time and amount, each row in the datasets correspond a transaction record on the blockchain. Some WalletIDs appear multiple times in the dataset, representing the user's multiple bitcoin exchanges. Table 3 shows a segment of the data recorded on 15 June 2013. There are 1048196 users in the dataset. So, clustering is difficult.

Table 3. Partial transaction records on MtĠox.

Wallet	Entry	Date	Operation	Amount
292938a9-ea37-4d58-a6c2-a7774159dbf7	7e1db835-a3f0-4527-a804-dd47e5a5e59c	2013/6/15 23:48	Withdraw	−0.9318619
07df3a31-4bfe-4178-a05c-5daab4e96575	6603e225-cb52-45e9-8190-9ac8dc74048f	2013/6/15 23:49	Withdraw	−2.622
2720c9d5-add9-4319-aa4e-ffd3a0ef3e48	853b7efe-b3dc-4cc5-a779-c23c6ab759a8	2013/6/16 0:04	Deposit	0.01298472

How to map user-address is described as follows [29]. Let T_trans represent the transaction time recorded by Mt.Gox, and T_block denote the block creation time. Let B be the block height. When the user deposits money into Mt.Gox at time T_trans, we find the block B whose creation time is T_trans. Due to the delay of the transaction record in the blockchain, we traverse block B∼B+36 to find the transaction whose amount uniquely matches to the amount recorded by Mt.Gox. If we find it, the output addresses of the transaction recorded on the blockchain are associated with the current user [9]. When the user withdraws the money from Mt.Gox, we traverse block B-6-36∼B-6-1 (A transaction is confirmed after 6 new blocks generated [30]) to map the input addresses with the user.

We choose the data from 1 May 2013 to 31 July 2013. Finally, 7945 unique (walletID, address) pairs are found. The partial results are shown in Table 4.

Table 4. Partial Results of (walletID, address) Pairs: Some walletIDs in MtĠox and Bitcoin addresses they used in blockchain.

walletID	Address
162aa442-0771-4bf7-a917-44b13506c139	1NYTFydxJFVEosz8M4KMQwWgmwPCTo6uVM
162aa442-0771-4bf7-a917-44b13506c139	1HwjJsntuJ8EU1r3xa3yZ8unFBmdR4BnS7
2663b417-a49b-4e34-a102-17f6cb885bda	1KmN99HPiYgVfvD1v648cccBbEkuRZgt19
203354fb-663b-4ee3-85e9-7d85c357927b	1FHQV8uGggQBnsE6XggP3c4iJpdjpKcE5u
203354fb-663b-4ee3-85e9-7d85c357927b	18C7SGRMNuCrmKypEnDzyi92QoyBzvwkMV

We arbitrarily extract two addresses from the above results to form address pairs marked as $(\boldsymbol{a_p}, \boldsymbol{a_q})$. If $\boldsymbol{a_p}$ and $\boldsymbol{a_q}$ have the same walletID, the label of $(\boldsymbol{a_p}, \boldsymbol{a_q})$ is 1. If not, the label is 0. There are then 5496694 negative samples and 13853 positive samples. Among them, the number of available samples of negative cases is much more than that of positive cases. In order to ensure the balance of the number of samples and improve model accuracy, the available samples of negative cases are randomly screened, so that the number of positive and negative cases are basically the same [17].

We use the API in [31] to find the transactions in which the Bitcoin addresses participate, and calculate the features' value of each unique address listed in Sect. 3.1. Then we combine the features of the two addresses in address pairs mentioned in Sect. 3.2, with the form: $(\boldsymbol{a_p}, \boldsymbol{a_q}, label) = (x_1, x_2, \ldots, x_i, \ldots, x_n, y_1, y_2, \ldots, y_i, \ldots, y_n, l)$, where l represents the label. Then, the shape of the final samples is $(27670, 37)$.

Table 5. The evaluation scores of machine learning models

Metrics	DNN	GBDT	XGBoost	LightGBM	Two-level learner
Time (µs)	368731	583201	497931	194901	1276033
Precision	0.5070	0.7940	0.7951	0.7870	**0.9603**
Recall	0.4763	0.8047	0.7855	0.7755	**0.9570**
F1	0.4921	0.7993	0.7903	0.7812	**0.9586**

4.2 Model Training

For all the models, we divide 80% of the samples into the training dataset and 20% as the testing dataset. 4-fold cross validation is employed in the training dataset [32]. The size of verification datasets is kept consistent with that of testing datasets for the purpose of gaining the best results. The verification datasets are used to verify whether the hyper-parameters of the model are tuned to be optimal. The optimal hyper-parameters and training datasets are used to train the model again, and then the model is utilized on the testing datasets.

After we get the samples in Sect. 4.1, we send them to the first-level learner. To prevent overfitting, in GBDT model, every iteration adopts 90% samples and considers up to 35 features when looking for the best split. For each tree in the GBDT model, the minimum number of samples required to split an internal node is 88. In XGBoost model, we extract 90% of the samples and 70% of the features randomly to train each boosting tree. In LightGBM model, we choose 90% of the samples in every 5 iteration and 60% of the features randomly in each iteration.

The DNN in the second-level learner uses Adam [33] as the optimizer. The purpose is to minimize the sum of categorical cross-entropy loss. The class labels are encoded as a one-hot vector. Finally, the learning procedure stops after around 35 epochs.

4.3 Experimental Results

Feature Analysis. A heat-map is employed to represent the Pearson correlation [34] between features. The feature_1 and feature_2 are used to represent the address i's features and address j's features, respectively, according to Sect. 4.1. As Fig. 3 shows, most Pearson correlation coefficient is around 0, which explains there are not too many features strongly correlated with each other. This is good from the point of view of feeding these features into our learning model, because this means that there is not much redundant or superfluous data in our datasets, and each feature carries some unique information. For example, the Pearson correlation coefficient between in_degree_1 and avg_val_1 is 0.01; they are therefore both important to our classification model.

The feature importance ranking in each basic tree boosting model is calculated by how many times a feature is used to separate decision trees. It is shown in Fig. 4. We combine addresses in the order $(\boldsymbol{a_i}, \boldsymbol{a_j})$, and the feature importance ranking of the three models is shown in Fig. 4(a), (b) and (c). As shown in the figures, these three models have the similar feature importance ranking. The features which have higher scores are related to time, value, amount and degree of the users' transaction. The result is consistent with the feature analysis in Sect. 3.1. We can then conclude that the addresses belonging to a user have the similar transaction characteristics, and we can safely utilize these features to correlate Bitcoin addresses.

Address Clustering. According to the experimental results of the classification, the addresses belonging to one user can be aggregated together. In the testing dataset, we get 1299 groups of addresses. The addresses in each group belong to the same user. Figure 5 shows the distribution of the number of addresses owned by the user. Most users have a small number of addresses. It illustrates that if we cluster the addresses directly, there are few samples in each category, and the information in each category is not sufficient. Clustering directly will thus be a complex task.

Performance Analysis. Three common and widely-used metrics, i.e., precision, recall and F1-measure are adopted to validate our proposed model. These metrics are formulated as follows: $precision = \frac{TP}{TP+FP}$, $recall = \frac{TP}{TP+FN}$, and $F_1 = \frac{2TP}{2TP+FP+FN}$, where TP and FP represent the number of items correctly and incorrectly labelled as belonging to the positive class, respectively. FN is the number of items incorrectly labelled as belonging to the negative class.

We use the unit of millisecond to measure the time of the model spending from training to prediction. It reflects the operational efficiency of each model. The two-level learner's time is the sum of the time spent by all the models involved in the two-level learners.

The different evaluation scores of machine learning models are shown in Table 5. GBDT, XGBoost and LightGBM as tree boosting models can handle all kinds of features well and have strong generalization ability (see the analysis

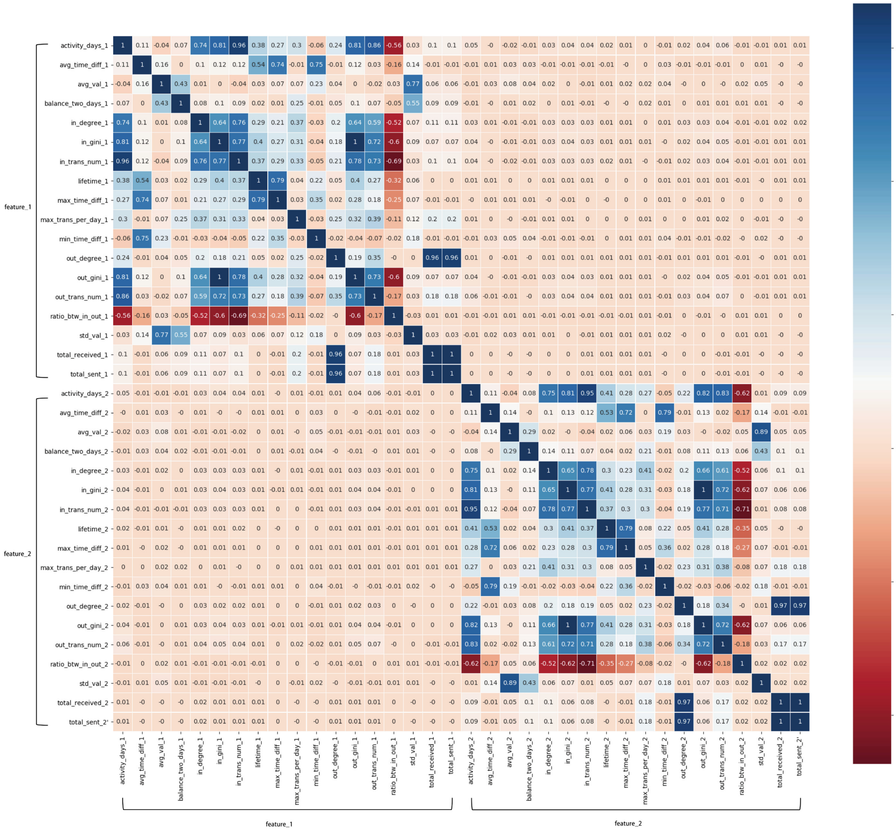
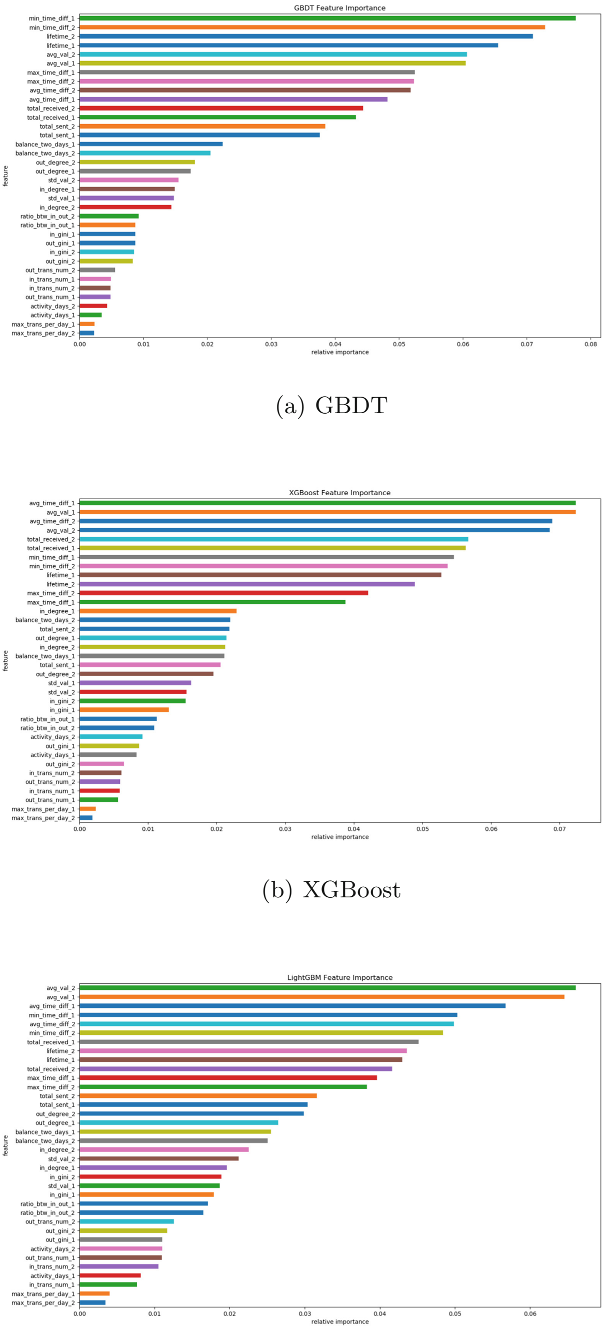


Fig. 3. Pearson Correlation Coefficient of Features: 1 denotes the total positive linear correlation, 0 means no linear correlation, and −1 represents the total negative linear correlation.

in Sect. 3.3). They perform better than the DNN model in precision, recall and F_1 scores. We can see that GBDT, XGBoost and LightGBM get similar precision, recall and F_1 scores. Because of the small number of features and samples, XGBoost and LightGBM cannot significantly improve precision and show advantages compared with GBDT. However, LightGBM is superior to XGBoost and GBDT in running speed due to its parallel optimization (see the discussions in Sect. 3.3). As to the two-level learner, we gain about 18% higher scores compared to other single models due to its smoothing nature and the ability to highlight each base model on its best performance cases (see Sect. 3.3 for details). It proves the effectiveness of adding the DNN layer after the three basic models and building the two-level learner model for Bitcoin address association.

In comparison with the existing works which use deep learning method in Sect. 2, where the precision is 0.766 and recall is 0.836, our model outperforms it by almost 20% in precision and 10% in recall. When using the heuristic methods



(a) GBDT

(b) XGBoost

(c) LightGBM

Fig. 4. Feature importance ranking of GBDT, XGBoost and LightGBM

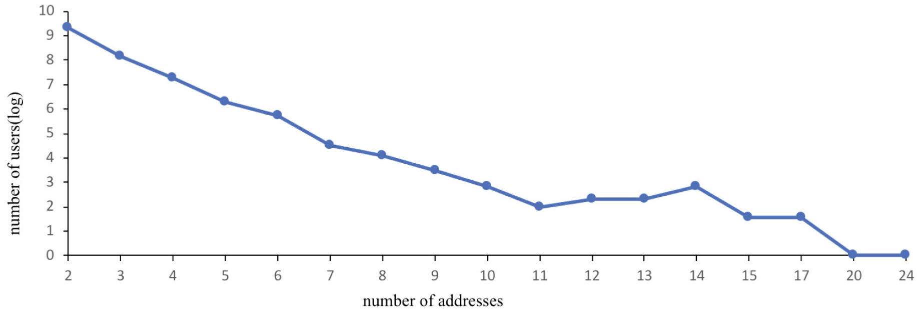


Fig. 5. The logistic function distribution of the number of addresses owned by the user: most users have a small number of addresses.

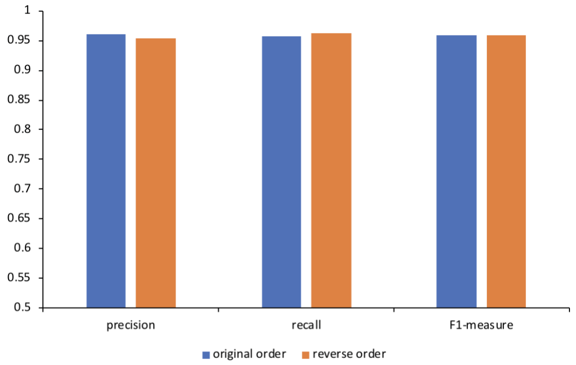


Fig. 6. The evaluation scores of different address combination orders: the blue bars represent original order and the red bars represent the reverse order. (Color figure online)

in Sect. 2, the mean recall is 0.709, our model outperforms it by almost 20% in recall.

For training sets, validation sets, and test sets, we generally use a consistent feature order as input. But in this experiment, we exchange the order of the two input addresses in the test datasets, that is, exchange the features' order. The evaluation scores of different address combination order are shown in Fig. 6. It shows that different orders result in the similar results. It illustrates that different orders do not have a significant impact on the classification of address pairs, which simplifies our address association work.

5 Conclusion

This paper has provided a Bitcoin address association method to perform de-anonymization, which transforms the clustering problem into a binary classification problem. The main idea of the method is to check if the two addresses belong to the same user and then cluster the addresses based on this check. The proposed method has combined both boosting tree models and deep learning models into a two-level learner to perform the classification. XGBoost, LightGBM and GBDT have been utilized as the first-level learner, and a three-layer deep neural network has been adopted as the second-level learner. By performance comparison, our method has performed more excellently than the ones which only employ one boosting method, such as XGBoost, LightGBM and GBDT, or only employs a deep neural network model, in terms of higher precision, recall and F1-measure scores. The research results in this paper can offer the suggestions and references for the investigation and tracking of illegal activities in Bitcoin systems and guide the blockchain system to study more secure and reliable encryption mechanisms.

References

1. ShenTu, Q.C., Yu, J.P.: Research on anonymization and de-anonymization in the bitcoin system. arXiv preprint arXiv:1510.07782 (2015)
2. Brenig, C., Accorsi, R., Müller, G.: Economic analysis of cryptocurrency backed money laundering. In: ECIS (2015)
3. Fanusie, Y., Robinson, T.: Bitcoin laundering: an analysis of illicit flows into digital currency services. Center on Sanctions & Illicit Finance memorandum, January 2018
4. Liao, K., Zhao, Z., et al.: Behind closed doors: measurement and analysis of CryptoLocker ransoms in bitcoin. In: 2016 APWG Symposium on Electronic Crime Research (eCrime), pp. 1–13. IEEE (2016)
5. Meiklejohn, S., Pomarole, M., et al.: A fistful of bitcoins: characterizing payments among men with no name. In: Proceedings of the 2013 Conference on Internet Measurement Conference, pp. 127–140. ACM (2013)
6. Fleder, M., Kester, M.S., Pillai, S.: Bitcoin transaction graph analysis. arXiv preprint arXiv:1502.01657 (2015)
7. Ermilov, D., Panov, M., Yanovich, Y.: Automatic bitcoin address clustering. In: 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 461–466. IEEE (2017)
8. Nick, J.D.: Data-driven de-anonymization in bitcoin. Master's thesis, ETH-Zürich (2015)
9. Reid, F., Harrigan, M.: An analysis of anonymity in the bitcoin system. In: Altshuler, Y., Elovici, Y., Cremers, A., Aharony, N., Pentland, A. (eds.) Security and privacy in social networks, pp. 197–223. Springer, New York (2013). https://doi.org/10.1007/978-1-4614-4139-7_10
10. Ron, D., Shamir, A.: How did dread pirate roberts acquire and protect his bitcoin wealth? In: Böhme, R., Brenner, M., Moore, T., Smith, M. (eds.) FC 2014. LNCS, vol. 8438, pp. 3–15. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44774-1_1
11. Ron, D., Shamir, A.: Quantitative analysis of the full bitcoin transaction graph. In: Sadeghi, A.-R. (ed.) FC 2013. LNCS, vol. 7859, pp. 6–24. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39884-1_2
12. Kaminsky, D.: Black ops of TCP/IP. Black Hat USA, p. 44 (2011)
13. Biryukov, A., Pustogarov, I.: Bitcoin over tor isn't a good idea. In: 2015 IEEE Symposium on Security and Privacy, pp. 122–134. IEEE (2015)

14. Mastan, I.D., Paul, S.: A new approach to deanonymization of *unreachable* bitcoin nodes. In: Capkun, S., Chow, S.S.M. (eds.) CANS 2017. LNCS, vol. 11261, pp. 277–298. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-02641-7_13
15. Shao, W., Li, H., Chen, M., Jia, C., Liu, C., Wang, Z.: Identifying bitcoin users using deep neural network. In: Vaidya, J., Li, J. (eds.) ICA3PP 2018. LNCS, vol. 11337, pp. 178–192. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-05063-4_15
16. Sanjaya, C., et al.: Revenue prediction using artificial neural network. In: 2010 Second International Conference on Advances in Computing, Control, and Telecommunication Technologies, pp. 97–99. IEEE (2010)
17. Bartoletti, M., et al.: Data mining for detecting Bitcoin Ponzi schemes. In: 2018 Crypto Valley Conference on Blockchain Technology (CVCBT), pp. 75–84. IEEE (2018)
18. Ghahramani, Z.: Unsupervised learning. In: Bousquet, O., von Luxburg, U., Rätsch, G. (eds.) ML -2003. LNCS (LNAI), vol. 3176, pp. 72–112. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-28650-9_5
19. Fan, X., Hongbo, X., Liang, Y.: A sock-puppet relation detection method on social network. J. Chin. Inf. Process. **28**(6), 162–168 (2014)
20. LeCun, Y., Bengio, Y.: Deep learning. Nature **521**(7553), 436 (2015)
21. Friedman, J.H.: Greedy function approximation: a gradient boosting machine. Ann. Stat. **29**(5), 1189–1232 (2001)
22. Chen, T., Guestrin, C.: XGBoost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794. ACM (2016)
23. Ke, G., Meng, Q., et al.: LightGBM: a highly efficient gradient boosting decision tree. In: Advances in Neural Information Processing Systems, pp. 3146–3154 (2017)
24. A guide to model stacking in practice (2016). http://blog.kaggle.com/2016/12/27/a-kagglers-guide-to-model-stacking-in-practice
25. Nair, V., Hinton, G.E.: Rectified linear units improve restricted Boltzmann machines. In: Proceedings of the 27th International Conference on Machine Learning (ICML-10), pp. 807–814 (2010)
26. Liu, W., Wen, Y., Yu, Z., Yang, M.: Large-margin softmax loss for convolutional neural networks. In: ICML, vol. 2 (2016)
27. mtgox2014leak. https://www.reddit.com/r/mtgoxAddresses/wiki/mtgox2014leak (2014)
28. Chen, W., Wu, J., Zheng, Z., Chen, C., Zhou, Y.: Market manipulation of bitcoin: evidence from mining the Mt. Gox transaction network. In: IEEE INFOCOM 2019-IEEE Conference on Computer Communications, pp. 964–972. IEEE (2019)
29. Xing, Y., Li, X., et al.: Research on de-anonymization techniques of bitcoin trading network. A Thesis Submitted to Southeast University For the Academic Degree of Master of Engineering, China (2017)
30. Nakamoto, S.: Bitcoin: a peer-to-peer electronic cash system. Consulted (2008)
31. Blockchain data API. https://www.blockchain.com/zh/api/blockchain_api (2017)
32. Arlot, S., Celisse, A., et al.: A survey of cross-validation procedures for model selection. Stat. Surv. **4**, 40–79 (2010)
33. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. Comput. Sci. (2014)
34. Pearson correlation coefficient. https://en.wikipedia.org/wiki/Pearson_correlation_coefficien (2019)