Identifying Illicit Addresses in Bitcoin Network

Yang Li[1,2], Yue Cai[1,2], Hao Tian[1,2], Gengsheng Xue[1,2], and Zibin Zheng[1,2(✉)]

[1] School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510275, China
{liyang99,caiy26,tianh23,xuegsh}@mail2.sysu.edu.cn
[2] National Engineering Research Center of Digital Life, Sun Yat-sen University, Guangzhou, China
zhzibin@mail.sysu.edu.cn

Abstract. Bitcoin has attracted a lot of attentions from both researchers and investors since it was first proposed in 2008. One of the key characteristics of Bitcoin is anonymity, which makes the Bitcoin market unregulated and a large number of criminal and illicit activities are associated with bitcoin transactions. Therefore, it's necessary to identify the illicit addresses in the Bitcoin network for safeguarding financial systems and protecting user's assets. To identify the illicit addresses in the Bitcoin network, first, we collect a large dataset of illicit addresses. The illicit addresses come mainly from some specific websites, public forums, and research papers. Second, we make a careful design of the features of illicit addresses. The features include basic features that refer to the related papers and the novel proposed features (topological features and temporal features). Third, we apply various machine learning algorithms (RF, SVM, XGB, ANN) to evaluate our features, which indicates that the proposed features are discriminating and robust. Besides, the paper discusses the class imbalance problem and achieves a better enhancement when using the cost-sensitive approach. Moreover, the paper proposes a model that incorporates LSTM into auto-encoder to generate temporal features. Results show that the generated features are helpful for the illicit addresses identification. Finally, the dataset and code are released in Github.

Keywords: Bitcoin · Illicit addresses · Machine learning · Auto-encoder · Topological features · Temporal features

1 Introduction

Bitcoin has attracted extensive attention from both investors and researchers since it was first proposed by Nakamoto [1] in 2008. It is the first open-source and widest spread digital cryptocurrency that has no central authority to control or manage its supply. Bitcoin works on the principle of a public decentralized ledger called blockchain [2,3], which is the core mechanism and provides security

Z. Zheng et al. (Eds.): BlockSys 2020, CCIS 1267, pp. 99–111, 2020.
https://doi.org/10.1007/978-981-15-9213-3_8

for the Bitcoin network. A blockchain consists of the longest series of blocks from the genesis block to the current block that is linked using cryptography. The process of new coins created is known as Bitcoin mining [4,5], which is to solve a computation problem.

One of the key characteristics of Bitcoin is the high anonymity it provides for its participants [6]. Bitcoin addresses are the only information used to send or receive Bitcoins for participants who do not need to provide any information on identification. Although address' information (all historical transactions and balances) can be obtained through the public decentralized ledger once the address is used, it is still impossible to de-anonymize it. Thus, there exist a wide range of crimes such as murders for hire, funding terrorism, drug, weapon, organ trafficking, Ponzi schemes, forgeries, unlawful gambling, money laundering, illegal mining, computer hacking, spreading ransomware and outright theft [7–9]. Therefore, identifying illicit addresses play a critical role in safeguarding financial systems, which is helpful for the Bitcoin ecosystem.

In addressing the aforementioned problems, we construct a large dataset that includes more than 20,000 illicit addresses and applies machine learning methods to identify them. More specifically, first, we collect the illicit addresses from various sources such as bitcoin websites, public forums, and some related papers. We also verify some addresses through open websites. Second, we not only collect the features which are used in related papers that are evidenced effectively but also propose two types of novel features (topological features and temporal features) in this paper. Last, we apply various machine learning algorithms (RF, SVM, XGB, ANN) to evaluate the proposed features and achieve good performance. Besides, the paper discusses the class imbalance problem and achieves a better enhancement using a cost-sensitive approach. Furthermore, the paper proposes a model that incorporates LSTM into auto-encoder to generate temporal features. Results show that the generated features are helpful for the illicit addresses identification.

In summary, our main contributions are:

- Dataset: a dataset of illicit addresses are collected from various source.
- Features: two types of novel features (topological features and temporal features) are proposed to identify illicit addresses.
- Algorithm: an auto-encoder with an LSTM model is proposed to generate new temporal features.
- Experiments: various and sufficient machine learning methods are used to identify illicit addresses and the class imbalance problem is discussed in this paper.

The rest of this paper is organized as follows. Section 2 investigates the related work of identifying illicit addresses. Section 3 illustrates our methodology of collecting illicit addresses dataset and constructing three types of features. Section 4 compares the effectiveness of various machine learning methods and discusses the class imbalance problem. Besides, we propose a model that incorporates LSTM into auto-encoder to generate new features to enhance the prediction. Finally, Sect. 5 draws some conclusions.

2 Related Work

Identifying illicit addresses play a critical role in safeguarding financial systems. Studies can be divided into two categories for identifying illicit addresses in the Bitcoin network. The first is to detect anomalous users and transactions. The second is to focus on specific illicit addresses such as scam, ransomware, Darknet market, Hack.

For identifying anomalous users and transactions in the Bitcoin network, [10] proposes three main social network techniques to detect potential anomalous users and transactions in the Bitcoin transaction network. [11] recently proposes unsupervised learning approaches to detect anomalies in the Bitcoin transaction network. [12] proposes a supervised classification model for detecting abnormality of Bitcoin network addresses. [13] presents graph convolutional networks for financial forensics on the Elliptic dataset. The dataset is also used in our paper.

For the scam identification, [14] applies data mining techniques to detect Bitcoin addresses related to Ponzi schemes. [15] proposes a novel methodology for HYIP (high yield investment programs) operators' Bitcoin addresses identification. For the ransomware identification, [16] proposes a network topology to measure and analyze ransomware in the Bitcoin network. For the Darknet market identification, [9] builds a dynamic research model to examine the evolution of Bitcoin and Darknet markets. However, the illicit addresses used in the above researches are small.

We collect a large illicit addresses' dataset. To the best of our knowledge, there exist no works that learn a recognized model for all types of illicit addresses on a large illicit addresses' dataset with advanced supervised learning methods.

3 Dataset Construction

3.1 Tag Collection

We develop a web crawler for public forums, user profiles (e.g., Bitcointalk.com, Twitter and Reddit) and darknet markets (e.g.., Silkroad, The Hub Marketplace and Alphabay) with some keywords (e.g., drug, arms, Ponzi, investment, ransomware, blackmail scam, sextortion, bitcoin tumbler, darknet market, ...). Especially, we crawl and filter bitcoin addresses of Bitcoinica Hack in https://bitcointalk.org/index.php?topic=576337. The data crawled from these sites is called crawled data. We also extend our search by considering all addresses listed on www.blockchain.com/btc/tags, a website that allows users to tag Bitcoin addresses. Most of the tagged addresses also contain a link to the website where they are mentioned and the description of tags. We filter the illicit addresses by the tags mentioned above. www.bitcoinabuse.com is a public database of bitcoin addresses used by hackers and criminals, which tracks bitcoin addresses used by ransomware, blackmailers, fraudsters, and so on. We download all the illicit addresses from it, addresses are also classified. www.bitcoinwhoswho.com is a website which provides all available information about a bitcoin address and it will report some Bitcoin Scams. Thus, we crawl some Bitcoin scam addresses

from it. The data tagged from these websites is called tagged data. Besides, we investigate some papers which are related to the identify illicit addresses. [14] releases a dataset of real-world Ponzi schemes. [17] releases the ransomware seed dataset. [13] releases an Elliptic dataset which includes 11,698 illicit addresses (scams, malware, terrorist organizations, ransomware, Ponzi schemes). However, they don't provide the specific labels of each illicit entity. The data collected from these papers is called paper data. Besides, we removed addresses without any transactions. These addresses may be used for scams or other illicit usages, but no one is fooled by them.

Overall, we find 24,720 illicit addresses which can be categorized as follows and the details we display in Table 1.

- **Ponzi scheme** is fraudulent investments that repay users with the funds invested by new users that join the scheme and implode when it is no longer possible to find new investments.
- **Ransomware** is spread to lock or encrypt the database, files, PC, or any electronic copy and demand ransoms in Bitcoin to enable access.
- **Blackmail** Bitcoin holder knowingly sends Bitcoin to criminals because of threatening or blackmail.
- **Darknet market** is a commercial website on the web that operates via darknets such as Tor or I2P. We collect the illicit addresses like arms trade, human trafficking, pornography and violence, drugs, etc.
- **Hack** wallets belonged to an exchange or a platform are hacked by outsiders, which led to the collapse of the exchange.

Table 1. Classes of illicit addresses

Class	Number	Source
Ponzi schemes	120	Tagged data, crawled data, papers data
Ransomeware	8979	Tagged data, crawled data, papers data
Blackmail	2884	Crawled data, tagged data
Darknet market	293	Crawled data, tagged data
Hack	406	Crawled data, tagged data
Unknown	11698	Only from paper [13]
Others	340	Tagged data
Total	24720	

3.2 Automatic Addresses Filtering

Some addresses may be inevitably misidentified as illicit addresses. We make an automatic address filtering, leveraging addresses clustering. Addresses clustering

is based on a heuristic [18]. It can be described as if two or more addresses are inputs of the same transaction with one output, then, all these addresses are controlled by the same user. This heuristic is expected to be accurate since Bitcoin clients do not provide support for different users to participate in a single transaction.

The considerations of filtering licit addresses can be summarized into two parts. First, addresses may mix with some normal addresses like exchanges, services since there exist multiple services for multi-input transactions nowadays. It's hard to label them as normal addresses or illicit addresses. Secondly, addresses may be normal scams that live a long time. Thus, it's also hard to label them. Therefore, we remove these uncertain addresses. Overall, we remove 523 illicit addresses whose multi-input addresses are more than 1000 from our collected dataset.

3.3 Discriminating Features Extraction

Here, we extract various features from illicit addresses, which are used to detect with supervised learning algorithms. Features can be roughly categorized as follows: 1) basic features, they are obtained from existing literatures; 2) topological features, they are extracted from the topological structure of transactions; and 3) temporal features, they are obtained from the change of balance of addresses. The following subsections provide more details on each type of feature.

Basic Features (BaF). The basic information of an address is used for feature construction. It includes the sum of all the inputs and outputs transferred to (resp. from) the address and final balance.

Besides, some basic features are obtained from [12,14,19]. The features in [14] are used for detecting Bitcoin Ponzi schemes. We select the lifetime of the address, the activity days, the maximum number of daily transactions to/from the address, the Gini coefficient of the values transferred to (resp. from) the address, the sum of all the values transferred to (resp. from) the address, the number of incoming (resp. outgoing) transactions which transfer money to (resp. from) the address, the ratio between incoming and outgoing transactions to/from the address, the average (resp. standard deviation) of the values transferred to/from the address, the minimum (resp. maximum, average) delay between the time as a part of our basic features. The features in [19] are used to identify what kind of services are operated by Bitcoin addresses. We select the frequency of transactions, payback ratio, the average numbers of inputs and outputs in the spent transactions as a part of our basic features. The features in [12] are used to classify Bitcoin addresses. We select the transaction moments which are proposed to encode temporal information as a part of basic features, the details are described in this paper.

Topological Features (ToF). Figure 1 shows that addresses with the same structure of transactions are labeled differently due to the property of input

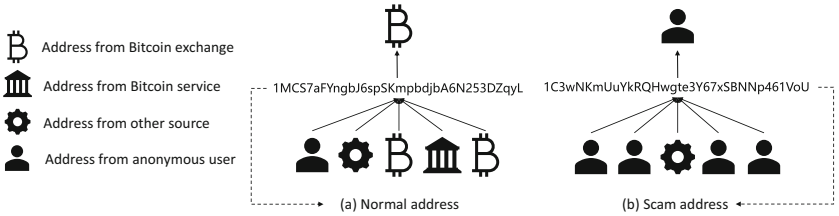

Fig. 1. A topological structure of a normal address and a scam address.

addresses and output addresses. More specifically, address a receives Bitcoins mainly from Bitcoin exchanges and services (provide bitcoin service for users) and sends to Bitcoin exchange. Thus, address a is a normal address with high probability and labeled as normal address since the KYC (Know Your Client, is the process of a business verifying the identity of its clients and assessing their suitability) is required by Bitcoin exchanges and some Bitcoin services. Instead, address b receives Bitcoins mainly from anonymous users and sends it to anonymous users. Thus, address b is an illicit address with some probability and labeled as a scam address. To solve this problem, we construct topological features to capture more information. At first, we characterize addresses into five categories (exchange, service, gambling, pool, and unknown user) followed by [20]. Then, each address will have a topological feature vector of length 10. For example, a topological feature vector is [1–4,7,10], the first five number represents that one input address comes from the exchange, three input addresses come from service, four input addresses come from gambling, seven input addresses come from the pool, and ten input addresses come from the unknown user. The second five number represents that four output addresses come from the exchange, two output addresses come from service, ten output addresses come from gambling, two output addresses come from the pool, and three output addresses come from the unknown user.

Temporal Features (TeF). Apart from the BaF and ToF, each address has different time distributions of transactions. In order to capture the temporal information, the time series of each address's balance (B) is constructed. More specifically, the address's balance will be updated and appended to the time series when it has a new transaction.

Figure 2 shows an example of a ransomeware (`12PEiX8JrYmpMRL6jkTK38pcDnq14NwVHB`) and Ponzi scheme address (`1Dgp5LqGZKWP7PrmxTG1Sitb88a16HzwCy`). It can be seen that the Ponzi address tends to receive bitcoins with the same amount every time and transfer a large number of bitcoins to other addresses at a time. To find some regular features, first, we apply the first difference method to the vector B and form a new vector C. Second, the mean and variance of C are calculated. Last, we define a time window of t, which is used to get a new vector V from C. The length of V is $len(C)/t$, and the values are the mean of C in the period of t. The mean and variance of V are

calculated as features. Here, t is set to $2, 4, 6, 8$ respectively. Besides, the main difference between the ransomware addresses and normal addresses is that the number of bitcoins received from normal addresses may have irregular decimals. However, ransomware addresses may receive the number of bitcoins with regular decimals or integer due to the value of bitcoins. Therefore, if $mean(C) = 0$ and $Round(sum(a), 2) = a$, we set 1 as the feature, otherwise, we set 0 as the feature.

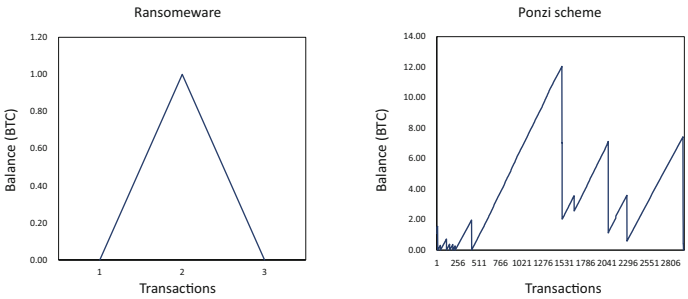

Fig. 2. An example of a ransomware address and a Ponzi scheme address.

4 Supervised Learning for Illicit Addresses Identifying

4.1 Data and Experimental Setup

The dataset includes 24,197 illicit addresses and other 1,209,850 licit addresses. The illicit addresses described in Sect. 3 are labeled as 1. The licit addresses sampled from WalletExplorer.com are labeled as 0. All 92 features described in Sect. 3 are used as model inputs. The imbalance ratio between two classes is 1:50 (1 illicit address every 50 instances of licit addresses). The main reason is that [14] proposes 1 Ponzi instance in every 200 instances of non-Ponzi and we expand to 1:50 in our illicit addresses compared to licit addresses because we have a lot of other types of illicit addresses. Besides, the dataset are divided into a training set (80% of the dataset), validate set (10% of the dataset), and testing set (10% of the dataset).

4.2 Classifiers and Evaluation Metrics

In this section, we evaluate the hand-crafted features (BaF, ToF, TeF) using several classic classifiers such as Random Forests (RF) [21], Support Vector Machine with RBF kernel (SVM) [22], XGBoost (XGB) [23], and Artificial neural networks (ANN) [24].

The implementation details are described as follows. We apply scikit-learn [25] which is a Python module for machine learning to SVM and RF. XGB

is implemented with the XGB python library which is open source at github. The artificial neural networks are implemented with Keras [26]. The architecture of artificial neural networks is composed of three fully-connected hidden layers and an output layer. Each hidden layer incorporates batch normalization and dropout regularization. Besides, we normalize the input data with a max-min method which changes the values of numeric columns to a common value between 0 and 1 since the neural networks are sensitive to the input data.

Precision, recall, and F1 score are used to evaluate the performance of the presented detection models. These metrics are capable to measure the imbalanced data.

4.3 Experimental Results

Table 2. Results of different classifiers

Method	Precision	Recall	F1
RF^{BaF}	0.9263	0.7069	0.8019
$RF^{BaF+ToF}$	0.9297	0.7231	0.8135
$RF^{BaF+ToF+TeF}$	**0.9355**	0.7293	0.8196
SVM^{BaF}	0.7813	0.6512	0.7103
$SVM^{BaF+ToF}$	0.8197	0.6742	0.7399
$SVM^{BaF+ToF+TeF}$	0.8355	0.6983	0.7608
ANN^{BaF}	0.7802	0.7844	0.7823
$ANN^{BaF+ToF}$	0.8712	0.7302	0.7945
$ANN^{BaF+ToF+TeF}$	0.8662	0.7750	0.8181
XGB^{BaF}	0.9049	0.8453	0.8741
$XGB^{BaF+ToF}$	0.9063	0.8529	0.8787
$XGB^{BaF+ToF+TeF}$	0.9100	**0.8540**	**0.8811**

Table 2 shows the detailed testing results in terms of precision, recall, and F1 score for the illicit class. Each model is executed with different input features. BaF refers to the basic features, ToF refers to the topological features, and TeF refers to the temporal features.

Note that XGB, RF outperform SVM and ANN, indicating the usefulness of the tree-based methods compared to other methods. Besides, XGB achieves the best F1 score and recall with all the three types of features. The main reason is that we can tune the parameter Scale_pos_weight of XGB which can suit for imbalanced data to obtain a good recall. Scale_pos_weight is the ratio of the number of negative class to the positive class.

Another insight from Table 2 is obtained from the comparison between features trained on the same model. For XGB, it can be seen that the enhanced feature (ToF, TeF) can improve the accuracy of the model only with BaF.

4.4 Class Imbalance Problem

The ratio of illicit addresses to licit addresses is 1:50 in the previous experiments [14]. However, the real-world distribution may be not equal to this specific ratio. In this section, the class imbalance problem is discussed. We change the imbalance ratio of training set and apply RF to evaluate. The ratio is set 1:200, 1:100, 1:50, 1:20, and 1:5 respectively.

Table 3. Results of different imbalance ratio

Imbalance ratio	Precision	Recall	F1
1:200	0.8448	0.4900	0.6200
1:100	0.8286	0.5800	0.6824
1:50	0.9355	0.7293	0.8196
1:20	0.9411	0.767	0.8452
1:5	**0.9567**	**0.84**	**0.8946**

Table 3 shows that the results are different with different imbalance ratio. The results achieve best in precision, recall and F1 score when the imbalance ratio is 1:5. Besides, the larger the imbalance ratio, the lower the recall.

To improve the results of class imbalance problem, we investigate it in data mining [27]. The solutions can be divided into sampling-based approaches and cost-sensitive approaches. Sampling-based approaches [28] construct balanced training set and adjusting the prior distribution for minority class (under sampling) or majority class (over sampling). Cost-sensitive approaches [29] is another type which takes the misclassification costs into consideration in the training phase. More specifically, the cost-sensitive approaches use a cost matrix to penalize different misclassification. For example, CM5 represents the cost of a false negative error is 5 times larger than the cost of a false positive error. In this paper, we only consider the cost-sensitive approaches because the sampling-based approaches will change the distribution of the training set, and the imbalance ratio is 1:50.

Table 4. Results of different cost matrix

Cost matrix	Precision	Recall	F1
CM5	**0.9038**	0.7737	0.8337
CM10	0.8768	0.8032	**0.8384**
CM20	0.8522	0.7976	0.8240
CM40	0.8181	**0.8327**	0.8253

The details of RF that applies cost-sensitive approach can refer to [30]. Table 4

shows that different cost matrices are used for RF model with the dataset. The best F1 score are achieved with the CM10 cost matrix. All the four cost matrices perform better than the origin RF with F1 score. However, it also indicates that the performance is sensitive to the cost matrix, the design of the cost matrix is important when we want to get a good result with cost-sensitive approaches.

4.5 Auto-encoder with LSTM

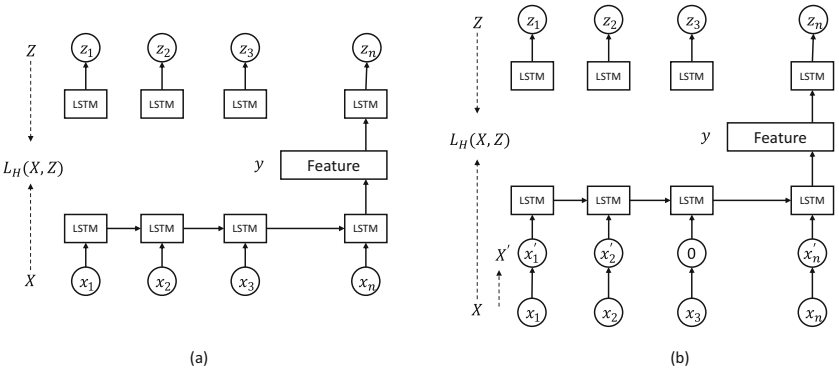

Fig. 3. Auto-encoder with LSTM.

The temporal features described in Sect. 3 are shallow and lack of hidden information since we only use a time window to obtain features. Thus, the regular rules or the trading behaviors in the transaction series are not included. In order to obtain more useful and discriminating temporal features, we apply an auto-encoder [31] method to generate it. An auto-encoder is a type of artificial neural network used to learn efficient features in an unsupervised process.

However, about 80% of all illicit addresses have less than 10 transactions. The main reason is that a lot of addresses are ransomware and blackmail, the extortioner (the address owner) may only use the address once when they blackmailed their users. For addresses with less than or more than 10 transactions, we adopt two different auto-encoders to obtain the hidden features. The first is an ordinary auto-encoder applied to addresses with less than 10 transactions, which is shown in Fig. 3(a), it includes three parts, encoder, decoder, and training. The main process of each part is described as follows.

- **Encoder.** The LSTM network [32] that transforms an input vector x into hidden representation y is called the encoder process. y is an encoder feature that is called generated features (GeF).
- **Decoder.** The hidden representation y is mapped back to a reconstructed dimensional vector z from a LSTM network. This mapping is called the decoder.

– **Training.** Autoencoder training consists of minimizing the reconstruction error, that is, carrying the following optimization:

$$\text{argmin} L_H(X, Z) \tag{1}$$

For addresses which have a large number of transactions (>10), we adopt a denoising auto-encoder with LSTM (L-DAE) to obtain the hidden features since a large number of transactions may contain noise. The architecture of denoising auto-encoder with LSTM is shown in Fig. 3(b) and the main process is the same as auto-encoder with LSTM. The main difference is that the denoising auto-encoder uses a denoising criterion, which can be described as each time a training example x is presented, a different corrupted version $\tilde{x}$ of it is generated according to $q_D(\tilde{x}|x)$. $q_D(\tilde{x}|x)$ is a stochastic mapping.

Table 5. Comparison of GeF and TeF

Method	Precision	Recall	F1
$\text{XGB}^{\text{BaF+ToF+TeF}}$	0.9100	0.8540	0.8811
$\text{XGB}^{\text{BaF+ToF+GeF}}$	0.9273	0.8735	0.8996
$\text{XGB}^{\text{BaF+ToF+TeF+GeF}}$	**0.9348**	**0.8805**	**0.9069**

Table 5 shows that the generated features are better than the hand-crafted TeF. A combination of four types of features (BaF, ToF, TeF, GeF) achieves the best with XGB model. Besides, we compare the features generated only by L-AE and the features generated by both L-AE and L-DAE. The results in Table 6 shows that both L-AE and L-DAE performs better than L-AE.

Table 6. Comparison of L-AE and L-DAE

Method	Precision	Recall	F1
XGB L-AE	0.9211	0.8725	0.8961
XGB L-AE + L-DAE	**0.9348**	**0.8805**	**0.9069**

5 Conclusions

In this paper, we collect a new dataset for identifying illicit addresses in the Bitcoin network. Illicit addresses include ransomware, Ponzi schemes, darknet market, blackmail, hack and some unknown addresses which are not specifically categorized. We introduce three types of features for the illicit addresses classification problem. The basic features are based on some previous works, which

are proved to be effective. The topological features contain extra information on addresses' inputs and outputs. The extra information can provide the trading behaviors and rules of the address. The temporal features are extracted from the change of addresses' balance. It can capture some regular patterns of addresses. Experimental results show that the performance of the proposed features and feature combinations improve the classification measurement. Besides, we make a sufficient discussion on the class imbalance problem because the amount of illicit addresses and licit addresses is a significant difference in the real world and we also apply cost-sensitive approaches to improve the results. Moreover, we provide a deep learning approach (auto-encoder with LSTM) to generate new temporal features and achieve better results than previous features.

Lastly, we hope the collected dataset and proposed methods in this field of identifying illicit addresses can attract more researchers and make the financial systems safer.

Acknowledgments. The work described in this paper was supported by the National Key Research and Development Program (2016YFB1000101), the National Natural Science Foundation of China (U1811462, 61722214) and the Key-Area Research and Development Program of Guangdong Province (2018B010109001).

References

1. Nakamoto, S., et al.: Bitcoin: A peer-to-peer electronic cash system (2008)
2. Zheng, Z., Xie, S., Dai, H., Chen, X., Wang, H.: An overview of blockchain technology: Architecture, consensus, and future trends. In: 2017 IEEE International Congress on Big Data (BigData Congress), pp. 557–564. IEEE (2017)
3. Zheng, Z., Xie, S., Dai, H.-N., Chen, X., Wang, H.: Blockchain challenges and opportunities: a survey. Int. J. Web Grid Serv. **14**(4), 352–375 (2018)
4. Chuen, D.L.K.: Handbook of Digital Currency: Bitcoin, Innovation, Financial Instruments, and Big Data. Academic Press, Cambridge (2015)
5. Eyal, I., Sirer, E.G.: Majority is not enough: bitcoin mining is vulnerable. Commun. ACM **61**(7), 95–102 (2018)
6. 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
7. Hurlburt, G.F., Bojanova, I.: Bitcoin: Benefit or curse? It Professional, **16**(3), 10–15 (2014)
8. Foley, S., Karlsen, J.R., Putninš, T.J.: Sex, drugs, and bitcoin: how much illegal activity is financed through cryptocurrencies? Rev. Financ. Stud. **32**(5), 1798–1853 (2019)
9. Janze, C.: Are cryptocurrencies criminals best friends? Examining the co-evolution of bitcoin and darknet markets (2017)
10. Pham, T., Lee, S.: Anomaly detection in bitcoin network using unsupervised learning methods. arXiv preprint arXiv:1611.03941 (2016)
11. Monamo, P., Marivate, V., Twala, B.: Unsupervised learning for robust bitcoin fraud detection. In: 2016 Information Security for South Africa (ISSA), pp. 129–134. IEEE (2016)

12. Lin, Y.-J., Wu, P.-W., Hsu, C.-H., Tu, I-P., Liao, S.: An evaluation of bitcoin address classification based on transaction history summarization. In: 2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), pp. 302–310. IEEE (2019)
13. Weber, M., et al.: Anti-money laundering in bitcoin: experimenting with graph convolutional networks for financial forensics. arXiv preprint arXiv:1908.02591 (2019)
14. Bartoletti, M., Pes, B., Serusi, S.: Data mining for detecting bitcoin Ponzi schemes. In: 2018 Crypto Valley Conference on Blockchain Technology (CVCBT), pp. 75–84. IEEE (2018)
15. Toyoda, K., Takis Mathiopoulos, P., Ohtsuki, T.: A novel methodology for hyip operators' bitcoin addresses identification. IEEE Access **7**, 74835–74848 (2019)
16. Liao, K., Zhao, Z., Doupé, A., Ahn, G.-J.: 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)
17. Paquet-Clouston, M., Haslhofer, B., Dupont, B.: Ransomware payments in the bitcoin ecosystem. J. Cybersecur. **5**(1), tyz003 (2019)
18. Androulaki, E., Karame, G.O., Roeschlin, M., Scherer, T., Capkun, S.: Evaluating user privacy in bitcoin. In: Sadeghi, A.-R. (ed.) FC 2013. LNCS, vol. 7859, pp. 34–51. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39884-1_4
19. Toyoda, K., Ohtsuki, T., Takis Mathiopoulos, P.: Multi-class bitcoin-enabled service identification based on transaction history summarization. In: 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), pp. 1153–1160. IEEE (2018)
20. Jourdan, M., Blandin, S., Wynter, L., Deshpande, P.: Characterizing entities in the bitcoin blockchain. In: 2018 IEEE International Conference on Data Mining Workshops (ICDMW), pp. 55–62. IEEE (2018)
21. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001)
22. Hearst, M.A., Dumais, S.T., Osuna, E., Platt, J., Scholkopf, B.: Support vector machines. IEEE Intell. Syst. Appl. **13**(4), 18–28 (1998)
23. 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)
24. Jain, A.K., Mao, J., Moidin Mohiuddin, K.: Artificial neural networks: a tutorial. Computer **29**(3), 31–44 (1996)
25. Pedregosa, F., et al.: Scikit-learn: machine learning in python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
26. Gulli, A., Pal, S.: Deep Learning with Keras. Packt Publishing Ltd. (2017)
27. Longadge, R., Dongre, S.: Class imbalance problem in data mining review. arXiv preprint arXiv:1305.1707 (2013)
28. Nguyen, G.H., Bouzerdoum, A., Phung, S.L.: Learning pattern classification tasks with imbalanced data sets. In Pattern recognition, IntechOpen (2009)
29. Sun, Y., et al.: Cost-sensitive boosting for classification of imbalanced data. Pattern Recogn. **40**(12), 3358–3378 (2007)
30. Bahnsen, A.C.: Ensembles of example-dependent cost-sensitive decision trees (2015)
31. Kramer, M.A.: Nonlinear principal component analysis using auto associative neural networks. AIChE J. **37**(2), 233–243 (1991)
32. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)