Learning To Deanonymize The Bitcoin Networks Using Neural Network

Yunfei Liu, Geyingjie Wen
Shanghai Jiao Tong University, China
{liuyunfei, geraint_j}@sjtu.edu.cn

***Abstract*—Bitcoin possesses strong security features which anonymizes user's identity to protect their private information. But some criminals utilize Bitcoin to do illegal activities, such as money-laundering. Therefore, it is necessary to learning to de-anonymize the bitcoin networks. Previous research has demonstrated that it is indeed possible to cluster together Bitcoin addresses and link such clusters to real-world identities. They use cluster method or network analysis method to deanonymize. In this report, we parser block and transaction details, and match bitcoin transaction with user exchange records which stored by a bitcoin company, then we can get each user's bitcoin addresses. Then use neural network to learn the feature of user bitcoin transaction, to achieve recognizing a new bitcoin address.**

***Index Terms*—Bicoin, Blockchain, de-anonymization, neural network**

I. Introduction

Bitcoin is a cryptocurrency and worldwide payment system, on which transactions are facilitated through a peer-to-peer network. Bitcoin is the earliest virtual currency developed in 2008 [1], ever since has attracted the attention of the research community due to its unique characteristics, such as decentralization and privacy. Bitcoin utilize a distributed shared-data database so that every Bitcoin client can verify and trace the transaction information. What's more, Bitcoin is pseudonymous, meaning that funds are not tied to real-world entities but rather bitcoin addresses. Owners of bitcoin addresses are not explicitly identified.

Bitcoin possesses strong security features which anonymizes user's identity to protect their private information. But some criminals utilize Bitcoin to do illegal activities, such as money-laundering. There have been articles and reports stating that Bitcoin has been used for terror financing, thefts, scams and ransomware. [2] [3] Therefore, in such case, uncovering the anonymity of the parties would be legally permissible and ethically desirable - but technically infeasible, according to popular belief about the robustness of the Bitcoin Blockchain anonymity.

In fact, many researchers have explored the limits of the anonymity of Bitcoin, previous research has demonstrated that it is indeed possible to cluster together Bitcoin addresses and link such clusters to real-world identities. Cluster Bitcoin address into distinct entities, which usually did the clustering by three kinds of heuristic method.

- Co-spend clustering: multiple input addresses of one transaction belong to the same entity.
- Intelligence-based clustering: Information is gathered from outside the blockchain, such as data leaks, court documents.
- Behavioural clustering: When user use a bitcoin wallet, which is hosted by a third party provider, Behavioural clustering can be used to cluster and relate the Bitcoin addresses to known hosted services or even to a specific wallet software.

Meiklejohn et al. [4] and BitIodine [5] made use of the above methods to cluster Bitcoin address. There also existed few works about deanonymizing Bitcoin addresses at IP level. Resolve the blockchain data to analyze Bitcoin from the point of Bitcoin address while simulate Bitcoin P2P protocol to evaluate Bitcoin from the point of IP address. Use blockchain data and network traffic Data to de-anonymize and extract all transaction and address from blocks to form a database. As for network, decoding the P2P protocol and crawl the bitcoin nodes' IP to count the active users. Then use the transaction packet as the feature to search in traffic data. We can get the bitcoin address and IP addresses. [7]

Throughout this paper we make the following contributions:

- Parser structurally block details, and build bitcoin transaction network for further analysis.
- Parser structurally the bitcoin exchange records between users and a bitcoin company. Then match each bitcoin exchange record with bitcoin transactoin stored in block chain. In this way, we can bind some bitcoin address and real world entity to achieve bitcoin de-anonymising work.
- We using generative adversarial network to generate some training set to solve unbalance class problem.
- We encode transactions feature of the corresponding real world entity, then we use RNN to learn the abstract feature and achieve classifying work.

II. System Overview

Figure 1 shows the design of our system. And there are three main steps in our method, which are respectively cluster, feature encoding, classification and prediction using neural network.

A. Cluster

In this step, we want to associate anonymous nodes in bitcoin networks with real world identities. So, we parser block files that store all bitcoin transaction details to get bitcoin address and the corresponding transaction features such

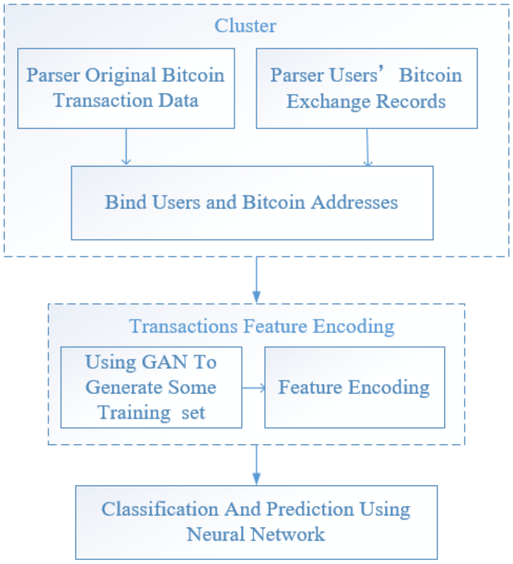
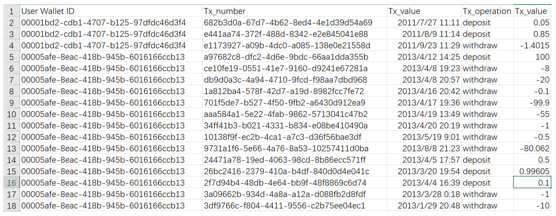

Cluster
Parser Original Bitcoin Transaction Data
Parser Users' Bitcoin Exchange Records
Bind Users and Bitcoin Addresses
Transactions Feature Encoding
Using GAN To Generate Some Training set
Feature Encoding
Classification And Prediction Using Neural Network

Fig. 1. Overview Of Our System

as transaction value and transaction time. Then we find a file named $btc\_xfer\_report.csv$ which records the bitcoin exchanges between users and bitcoin company Mt.Gox from GitHub. [8] The bitcoin company Mt.Gox was attacked by hackers in 2014, which divulged the recorded data. This file contains users' wallet ID, bitcoin exchange time and values, as shown in Figure 2. We match bitcoin exchange records with bitcoin transactions stored in blockchain, so that we can know the bitcoin addresses of each user.



1	User Wallet ID	Tx_number	Tx_value	Tx_operation	Tx_value
2	00001bd2-cdb1-4707-b125-97dfdc46d3f4	682b3d0a-67d7-4b62-8ed4-4e1d39d54a69	2011/7/27 11:11	deposit	0.05
3	00001bd2-cdb1-4707-b125-97dfdc46d3f4	e441aa74-372f-488d-8342-e2e845041e88	2011/8/9 11:14	deposit	0.85
4	00001bd2-cdb1-4707-b125-97dfdc46d3f4	e1173927-a09b-4dc0-a085-138e0e21558d	2011/9/23 11:29	withdraw	-1.4015
5	00005afe-8eac-418b-945b-6016166ccb13	a97682c8-dfc2-4d6e-9bdc-66aa1dda355b	2013/4/12 14:25	deposit	100
6	00005afe-8eac-418b-945b-6016166ccb13	ce10fe19-0551-41e7-9160-d9241e67281a	2013/4/8 19:23	withdraw	-8
7	00005afe-8eac-418b-945b-6016166ccb13	db9d0a3c-4a94-4710-9fcd-f98aa7dbd968	2013/4/8 20:57	withdraw	-20
8	00005afe-8eac-418b-945b-6016166ccb13	1a812ba4-578f-42d7-a19d-8982fcc7fe72	2013/4/16 20:42	withdraw	-0.1
9	00005afe-8eac-418b-945b-6016166ccb13	701f5de7-b527-4f50-9fb2-a6430d912ea9	2013/4/17 19:36	withdraw	-99.9
10	00005afe-8eac-418b-945b-6016166ccb13	aaa584a1-5e22-4fab-9862-5713041c47b2	2013/4/19 13:49	withdraw	-55
11	00005afe-8eac-418b-945b-6016166ccb13	34ff41b3-b021-4331-b834-e08be410490a	2013/4/20 20:19	withdraw	-1
12	00005afe-8eac-418b-945b-6016166ccb13	10138f9f-ec2b-4ca1-a7c3-d36f56bae3df	2013/5/19 9:01	withdraw	-0.5
13	00005afe-8eac-418b-945b-6016166ccb13	9731a1f6-5e66-4a76-8a53-10257411d0ba	2013/8/8 21:23	withdraw	-80.062
14	00005afe-8eac-418b-945b-6016166ccb13	24471a78-19ed-4063-98cd-8b86ecc571ff	2013/4/5 17:57	deposit	0.5
15	00005afe-8eac-418b-945b-6016166ccb13	26bc2416-2379-410a-b4df-840d0d4e041c	2013/3/20 19:54	deposit	0.99605
16	00005afe-8eac-418b-945b-6016166ccb13	2f7d94b4-48db-4e64-bb9f-48f8869c6d74	2013/4/4 16:39	deposit	0.1
17	00005afe-8eac-418b-945b-6016166ccb13	3a09662b-934d-4a8a-a12a-d088fb2d8fdf	2013/3/28 0:18	withdraw	-1
18	00005afe-8eac-418b-945b-6016166ccb13	3df9766c-f804-4411-9556-c2b75ee04ec1	2013/1/29 20:48	withdraw	-10

Fig. 2. Part Of File $btc\_xfer\_report.csv$

B. *Feature encoding*

In this step, we get each user's transaction features, including transaction time, sent bitcoin values and received bitcoin values. But we find some users have lots of bitcoin addresses (more than 100) and have made many bitcoin transactions. And some users only have less than 10 bitcoin address and the number of transactions is also less. We want to use those samples to train neural network, so that we can predict whom a strange bitcoin address belongs to. And unbalanced class in training set will lead our neural network model fail. So, we firstly handling training samples imbalance problem, then encoding the transaction feature of each user as the input of neural network.

C. *Classification and prediction using neural network*

Since the features we encoded in the last step are all sorted by time, so we prefer using a neural network that can deal with a temporal sequence, and Recurrent Neural Networks(RNN) will be a good choice for this work. RNNs are networks with loops in them, allowing information to persist, so they can learn the relationship about time between the events that happened in this temporal sequence. We can use Recurrent Neural Networks to train the data set and gain a model about the features of a user as the classification and judge whether a new bitcoin address belongs to any existed wallet ID.

III. Details Of Our System

A. *Cluster*

1) *Build bitcoin transaction network:*

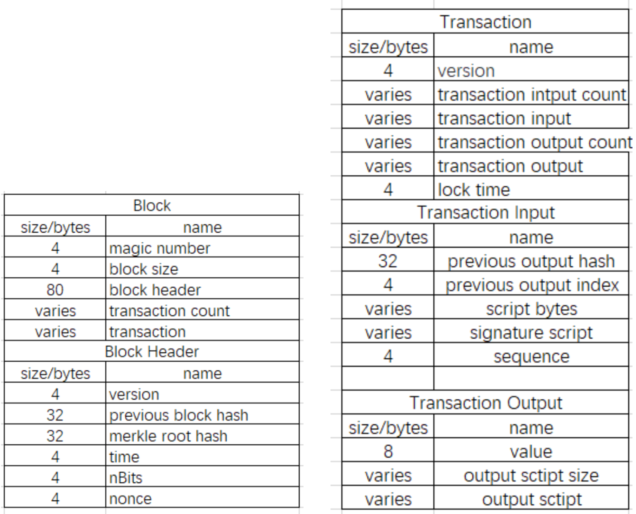
In the bitcoin transaction network, nodes are bitcoin addresses, and edges are bitcoin transactions. Since they are both recorded in bitcoin blockchain, so we need to get blockchain data. And this can be done by download a complete bitcoin client, and blk.dat hex files store all block information. We parse these hex files structurally and extract valid block and transaction information. In our work, we want to build bitcoin transaction network and match bitcoin exchange records between user and a bitcoin company with bitcoin transactions stored in blockchain. Thus, we need to get block timestamps, every transaction detailed information including input address, output address, transaction values and transaction time.

The structure of block and block header is shown in Figure 3(a). To get block timestamps, we need to get detailed information of block header and parser the four bytes timestamps. The structure of transaction in one block is shown in Figure3(b). In fact we can parser output script to get output bitcoin address, but it is difficult to parser input script to get input address, it is because that input script formats are different due to different kinds of transaction. A feasible method is to find previous output hash of this transaction input, and parser previous transaction's corresponding output script to get bitcoin address. In our work, we use a more convenient method to achieve it. We firstly get all transactions stored in one block, then use twice SHA-256 algorithm to calculate each transaction hash value, then we crawl each transaction details from bitcoin browser website using transaction hash value.

Through the above steps, we can get nodes and edges information of bitcoin transaction networks, then we can build this network which is shown as Figure4. We further can use this transaction network to match users' bitcoin exchange records. Since the bitcoin exchange is completed by the user and the bitcoin company, if there are several bitcoin addresses in the transaction inputs, we can think those input addresses belong to one user. It is also true for the output addresses.

2) *Bind user wallet ID and bitcoin addresses:*

From Figure2, we can see that there is no bitcoin address for each exchange record. Since we want to bind user wallet ID and bitcoin addresses, we design a match method based on exchange time and exchange value to find the corresponding



Block	
size/bytes	name
4	magic number
4	block size
80	block header
varies	transaction count
varies	transaction
Block Header	
size/bytes	name
4	version
32	previous block hash
32	merkle root hash
4	time
4	nBits
4	nonce

(a) The Strcuture Of Block And Block Header

Transaction	
size/bytes	name
4	version
varies	transaction intput count
varies	transaction input
varies	transaction output count
varies	transaction output
4	lock time
Transaction Input	
size/bytes	name
32	previous output hash
4	previous output index
varies	script bytes
varies	signature script
4	sequence
Transaction Output	
size/bytes	name
8	value
varies	output sctipt size
varies	output sctipt

(b) The Strcuture Of Transaction

Fig. 3. The Structure Of Block

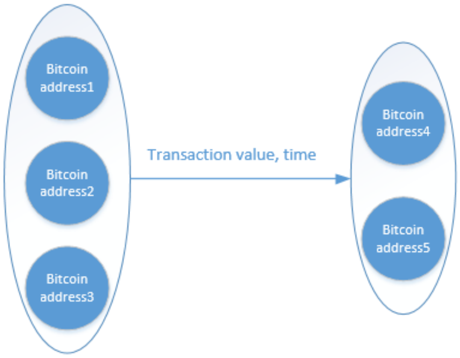


Fig. 4. A Bitcoin Transaction Network Diagram

bitcoin transaction of each exchange record. In the meanwhile, we notice that there are two exchange operations which are respectively deposit and withdraw operation. For the two different exchange operations, the corresponding matching criteria are also slightly different. The detailed matching criteria are described below.

For 'deposit' operation, it means user sent some bitcoin to the bitcoin exchange company. The recorded exchange time will be later than bitcoin transaction time. We find the first block whose timestamp is larger than exchange time. Then traverse transactions in this block and find the transaction whose transaction value is the same as exchange value. This transaction's output bitcoin addresses belong to this user. If failed, we search serval blocks forward.

For 'withdraw' operation, it means the bitcoin exchange company sent some bitcoin to this user. The bitcoin transaction time is later than exchange time. The match rules are similar with 'deposit' operation match rules. But if we fail, we will search serval blocks backward.

Using the above matching method, we can bind user wallet ID and bitcoin addresses. So that some anonymous bitcoin network nodes can be uncovered, just as Figure5 shown.

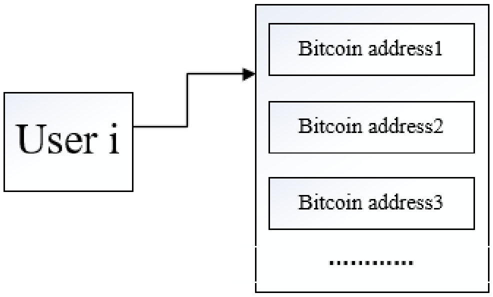


Fig. 5. Binding User And Bitcoin Address

3) Using K-means to cluster: Previous research has demonstrated that it is indeed possible to cluster together bitcoin address. [4] And we use cluster algorithm to compare with our matching method. We have matched bitcoin exchange records between user and a bitcoin company with bitcoin transactions stored in blockchain, then we extract the input addresses, output addresses, transaction value and transaction time of each transaction to form a feature vector, then use K-mean cluster algorithm for cluster analysis. K-means clustering is a method of vector quantization, originally from signal processing, that is popular for cluster analysis in data mining. K-means clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. By this way, we can aggregate similarly characterized transactions in one cluster. Since Each transaction occurs between the user and the company, and we can see a cluster as a user. Then we compare the cluster results with that based match rules to prove the effectiveness of our method.

B. Feature encoding

1) Handling class unbalance problem: It is normal that different users have different number of bitcoin addresses. We want to extract every bitcoin address?s behavioral characteristic of one user. In this situation, the user is the label of those bitcoin addresses? feature. Thus, our dataset is class imbalance. There are some methods to handle class imbalance problem, such as collecting more data, changing performance metric like confusion matrix, recall or F1 score, resampling our dataset including under-sampling and up-sampling, generating synthetic samples [6]. Thus, we think GAN is good way to generate some data similar to training dataset characteristics. But in our work, we do not have time to achieve it, so we analysis the number of bitcoin addresses of each user, we find less than ten users have more than 50 bitcoin addresses, most user have less than 10 bitcoin addresses. So we just delete some features of users who have more than 50 bitcoin address, and extract about 5 feature vectors from each user as training data.

2) encode feature:

Through the cluster work, we know the bitcoin addresses of each user. Then we want to learn the behavioral characteristics of each user's bitcoin transaction. First, since we have got the relationship of n user's wallet ID and k bitcoin addresses, we can get a $k \times n$ matrix which represents whether a bitcoin address belongs to a user, certainly it is one-hot. Then to quantify the daily behavior of each user, we choose one year(2011) and count daily transaction amount of each bitcoin address every day. We should notice that after a trade is initiated by a user and before it is recorded by blockchain, it must be accepted by at least 6 blocks. So target each trade to one day but not an accurate timestamp is enough. And to simplify the model, we can add up all the trades happened on a certain day. If the bitcoin address is input of one transaction, the value will be a negative number. If it is output of one transaction, the value will be a positive number. Then for this k bitcoin addresses, we can get a $k \times 365$ matrix, which represents the transaction amount of each bitcoin addresses on every day in 2011. With these features, now we can use neural network to classify a new bitcoin address.

C. Classification and prediction using neural network

Since we have finished encoding features, now we can using these data set and Recurrent Neural Networks to train a model as a classifier. Although RNNs can use previous information inform the understanding of the present information, as that gap grows, RNNs become unable to learn to connect the information. Long Short Term Memory networks are a special kind of RNN, capable of learning long-term dependencies. LSTMs are explicitly designed to avoid the long-term dependency problem. They contain the same model as standard RNNs but the module has a different structure, and they have three gates, 'forget gate', 'input gate' and 'output gate' to protect and control the states, just as Figure6 shown.

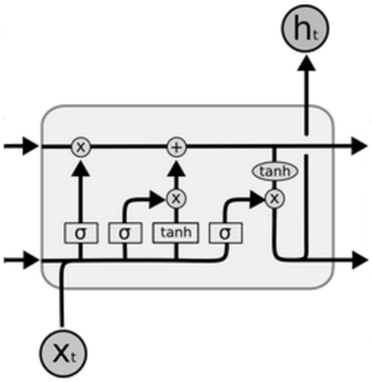


Fig. 6. LSTM module

In this work, we actually implement this lstm cell by a built-in cell that TensorFlow provides to us. The input part contains the features that we encoded, and to feed them into LSTMs, we set time step = 7 and num input = 53, since one year contains 53 weeks and we thought the behavior of a person will be regular on a week. And the output is a model that can classify whether a new address with its information belongs to an existed wallet ID or not.

IV. Experiments And Results

A. Cluster Results

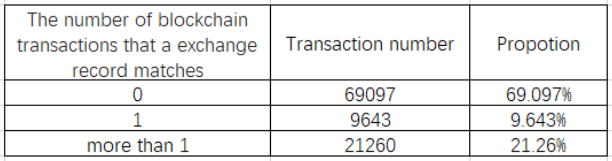
1) The detailed information of the block:

We structurally parser hex file blk.dat which stores blocks information by means of the code from GitHub. [9] Figure7 shows the parsed block detailed information where transaction hash value is not stored in block. In our work, we only need the block timestamps and transaction hash. We need to know that current block hash and current transaction hash are not stored in block, we use SHA-256 algorithm to generate them. So that we can get the block details from block browser website by the block hash to prove that the block information we have parsed out is accurate.

2) Bind user wallet ID and bitcoin address:

The file $btc\_xfer\_report.csv$ records about more than 3 million the bitcoin exchange between 2011 year and 2013 year. And we use the 100,000 records of 2011 year and match them with bitcoin transactions in bitcoin network. Figure8 shows the match results. If the match result is 0, it means that we can not find the corresponding blockchain transaction for the user bitcoin exchange record in our preset time frame. And only when the number of blockchain transaction that an exchange record matches is equal to 1, we think the match is successful.

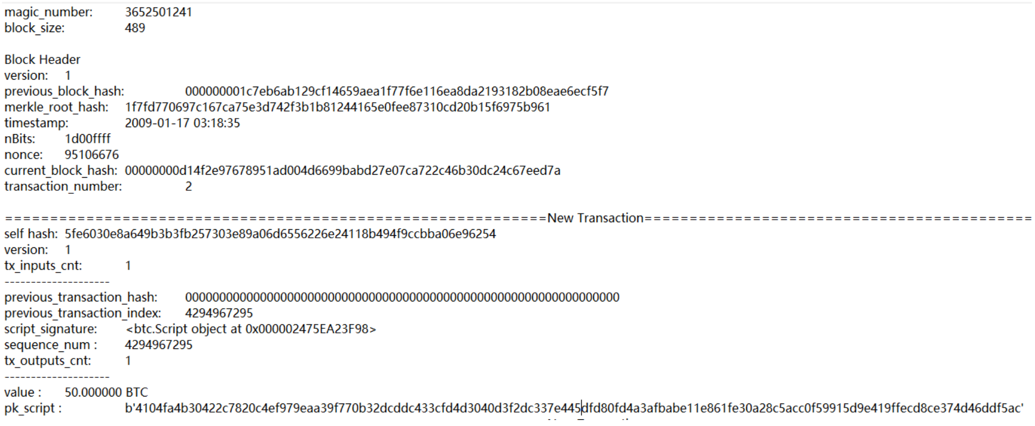
We can find the match success rate is not high, about 10%. This because that we only have the information of exchange value and exchange time, and the exchange time is not strictly equal to blockchain transaction time and there exists time delay. In our preset time frame, there may be lots of transactions with the same transaction value, which leads to multiple matches. Zero match rate is up to 69.1%, if we increase the preset time range, this rate can decrease a lot. However, in the meanwhile, the multiple matches rate will increase correspondingly a lot, which can not truly increase match success rate.



The number of blockchain transactions that a exchange record matches	Transaction number	Propotion
0	69097	69.097%
1	9643	9.643%
more than 1	21260	21.26%

Fig. 8. The Match Results

For the 9643 successfully matched exchange records, we know the corresponding bitcoin transaction hash, then we parser the bitcoin blockchain browser website http://www.qukuai.com/search/zh-CN to get transaction details, including transaction address, transaction values and transaction time. Then bind the user wallet ID and bitcoin address by the match method. Here we take the wallet ID



```
magic_number:      3652501241
block_size:        489

Block Header
version:    1
previous_block_hash:           000000001c7eb6ab129cf14659aea1f77f6e116ea8da2193182b08eae6ecf5f7
merkle_root_hash:   1f7fd770697c167ca75e3d742f3b1b81244165e0fee87310cd20b15f6975b961
timestamp:          2009-01-17 03:18:35
nBits:      1d00ffff
nonce:      95106676
current_block_hash: 00000000d14f2e97678951ad004d6699babd27e07ca722c46b30dc24c67eed7a
transaction_number:            2

============================================================New Transaction===========================================
self hash:  5fe6030e8a649b3b3fb257303e89a06d6556226e24118b494f9ccbba06e96254
version:    1
tx_inputs_cnt:       1
--------------------
previous_transaction_hash:     0000000000000000000000000000000000000000000000000000000000000000
previous_transaction_index:    4294967295
script_signature:     <btc.Script object at 0x000002475EA23F98>
sequence_num :        4294967295
tx_outputs_cnt:       1
--------------------
value :     50.000000 BTC
pk_script :           b'4104fa4b30422c7820c4ef979eaa39f770b32dcddc433cfd4d3040d3f2dc337e445dfd80fd4a3afbabe11e861fe30a28c5acc0f59915d9e419ffecd8ce374d46ddf5ac'
```

Fig. 7. The Parsed Block Details

5642ccb5-bdb3-4f9e-8229-fc5a72381010 as an example.This wallet ID has eight bitcoin exchange records in the 100,000 records of 2011 year, we find three bitcoin address from the corresponding eight bitcoin transactions, which are respectively 18uUotMFLADxWXGSasZqv2YkNeciLSGN6k, 1LX4e4TVtuLwW251G3Gd7uyZSk1M6YenCb, 1MWAT64zY4nrQaD4oVbLCgewcVb4vmDRmo.

We also count the number of addresses of each user wallet ID. The 9643 successfully matched exchange records contain 3347 different user wallet ID, and most user wallets contain less than 10 bitcoin address, which we think is the usage situation of normal users. Some user may have more than 100 bitcoin addresses, we guess they may be some bitcoin company or institute. The Figure9 shows the probability distribution of the number of bicoin addresses in one wallet, which is in line with the long tail distribution. [10]

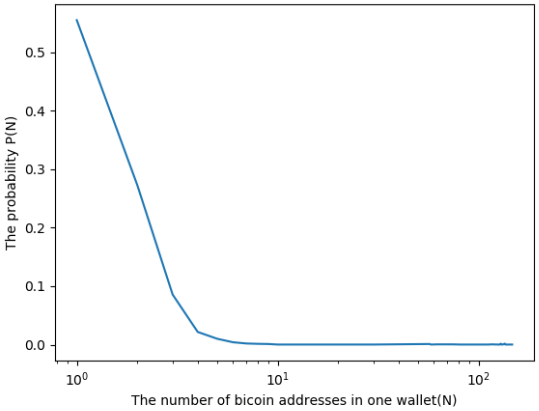


Fig. 9. The Probability Distribution Of The Number Of Bicoin Addresses In One Wallet

3) The results of k-means cluster:

We using the 9643 bitcoin transaction to extract feature vector, including transaction input address, output address, transaction time and transaction value. Then we use k-means with Euclidean distance measurement to cluster those transaction. We first decide the number of k, Figure10 shows the relationship between k-means cost and the value of k. As the results shown, when the value of k increases to 3000, the cost of k-means cluster drops about eight times. And when k is greater than 4000, the cost of k-means is low and tends to be stable. Thus, the value of k should be between 3000 and 4000. And k represents the number of different users in our transaction sets, and we have known that the 9643 matched successfully bitcoin transaction involves 3347 different user, which prove the effectiveness of k-means algorithm.

Then we set k as 3347, and we can use k-means to divide different transaction with the similar features to one cluster.

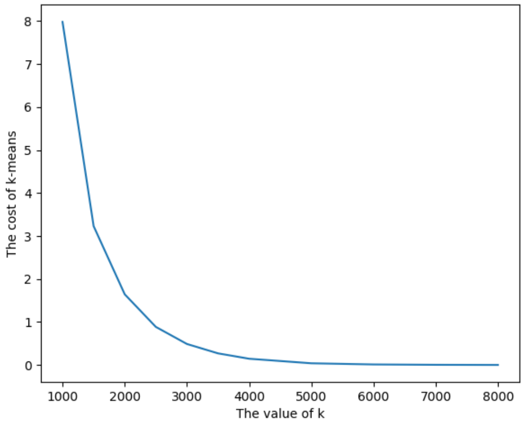


Fig. 10. The Influence Of k Value In K-means Algorithm

B. The results of classification and prediction

We using 404 samples from 100 wallet ID to train a network, the data set is 404×365, which means the trade value of a bitcoin address on a day in 2011, and the label set is 404×100, which means the wallet ID of a bitcoin address, obviously, it is one-hot. Then we mess up the order of the sets, and divide them into two parts, train set contains 360 bitcoin address and test set contains 44 bitcoin address.

After 20000 steps training, finally we get the model and the test data set tells us that we have a 47.83% accuracy rate.

The experimental results show that our method only get a low accuracy rate with current data set. We thought there is two main reasons. First, we only encoded several features of a bitcoin address's trade, and there can be more feature like the bitcoin address it traded with, since if two bitcoin addresses trade with one bitcoin address frequently may means that they belongs to one user(one wallet ID). Second, the dataset may not accurate enough and we need more data.

V. Conclusion

Our classification accuracy rate is not high, we think there are mainly the following reasons.

- Training dataset is not good enough. When we build training dataset, we bind user and bitcoin addresses. Just as the above analysis, our matching success rate is not high, which can lead to lose many transaction information of each user. Meanwhile, $btc\_xfer\_report.csv$ stores about 3 million bitcoin exchange records happened between real world users and bitcoin company, but we only use 100,000 records of them. So our training dataset may be have lost some vital feature which can classify different users.
- Feature encoding method needs improvement. When we encoding feature of each bitcoin address, we want to take advantage of the time sequence in which transaction occur. So we sorting transaction value into a sequence in chronological order of occurrence, which means that we have extract the time and value of one transaction. In fact, the transaction output addresses or output addresses is still important for studying a bitcoin address transaction feature. If study more deeply, we can extract more feature for each bitcoin address transaction.
- Neural network needs to train better. Neural network training work such as adjust parameters is very complex, and we need to put more effort to train neural network.

All in all, we roughly complete the de-anonymization of bitcoin. But we still need to do more work to improve our system and achieve a better accuracy rate.

VI. The division of labour

Liu yunfei: Cluster and transaction network analysis, feature encoding.

Wen Geyingjie: Feature encoding, using RNN to learn feature of transaction and achieve classification.

References

[1] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system[J]. 2008.

[2] Martin J. Drugs on the dark net: How cryptomarkets are transforming the global trade in illicit drugs[M]. Springer, 2014.

[3] Van Hout M C, Bingham T. 'Silk Road', the virtual drug marketplace: A single case study of user experiences[J]. International Journal of Drug Policy, 2013, 24(5): 385-391.

[4] Meiklejohn S, Pomarole M, Jordan G, et al. A fistful of bitcoins: characterizing payments among men with no names[C]//Proceedings of the 2013 conference on Internet measurement conference. ACM, 2013: 127-140.

[5] Spagnuolo M, Maggi F, Zanero S. Bitiodine: Extracting intelligence from the bitcoin network[C]//International Conference on Financial Cryptography and Data Security. Springer, Berlin, Heidelberg, 2014: 457-468.

[6] Haixiang G, Yijing L, Shang J, et al. Learning from class-imbalanced data: Review of methods and applications[J]. Expert Systems with Applications, 2017, 73: 220-239.

[7] Zhu, Jiawei, Peipeng Liu, and Longtao He. "Mining Information on Bitcoin Network Data."Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), 2017 IEEE International Conference on. IEEE, 2017.

[8] https://github.com/shemnon/GoxCalc/tree/master/src/main/resources

[9] https://github.com/huangsuoyuan/btc_parser

[10] https://en.wikipedia.org/wiki/Long_tail