Imperial College London

Department of Computing

How'd You Get So Rich? Behind the World's Richest Bitcoin Addresses

A Design of Bitcoin Tracing Software

By

Zhiping Huang (zh1516)

Submitted in partial fulfilment of the requirements for the MSc Degree in Computing of Imperial College London

August 2022

Acknowledgement

I would like to thank Professor William Knottenbelt for supervising me in my MSc project. Also, I would like to thank Aleš Janda, the author of Walletexplorer.com, for creating the website and offering help during my project.

Abstract

Bitcoin is a decentralised cryptocurrency introduced in 2008, it was designed to be intangible and anonymous. After the creation of bitcoin, it has attracted more and more attentions particularly in recent years. Due to the characteristics of cryptocurrency, it provides an ideal channel for illegal activities such as money laundering. In order to check the legality and composition of bitcoin funds present in a given bitcoin address, this project proposed and implemented a software designed using Python language. By acquiring bitcoin transaction data from available online services, it was able to construct a transaction network based on the unspent transaction output of the given address. From the constructed transaction network, collection of source addresses and calculation of the corresponding contribution of each address is achieved. This project also trained a bitcoin address classifier based on artificial neural network, this classifier was implemented in the bitcoin tracing software for classification of the collected source addresses. It was demonstrated that not only the software can be used to analyse the composition of bitcoin in a given address, but also it can be used to analyse the flow of bitcoin before transferring to the given address. It is believed that the software was robust and scalable. This constructed transaction network can be tailored with different parameters to keep up with the user's demands, extra tunning mechanisms can be easily added. The software can also be used to analyse all the addresses related to the given address, i.e in the same bitcoin wallet.

Contents

1 Introduction 1

2 Background and Related work 5

2.1 How bitcoin works 5

2.2 Web scraping 10

2.3 Related Work 11

3 Software Design and Implementation 13

3.1 Transaction Network 14

3.2 Bitcoin Address Classifier 18

3.3 Result Processing 20

4 Evaluation 21

4.1 Transaction Network 23

4.2 Bitcoin Address Classifier 25

5 Conclusions 28

6 Bibliography 29

7 Appendices 32

8 User guide 33

1 Introduction

Bitcoin is a decentralised cryptocurrency introduced in 2008 by Satoshi Nakamoto (1), it was designed to be intangible and anonymous based on a peer-to-peer network. It has attracted many attentions since it was introduced and has become increasingly popular in recent years. Bitcoins can be traded via online exchanges using various currencies. At the beginning, the price of one bitcoin is barely above zero USD, then the price gradually increases over time. The exchange rate firstly peaked at 2017, reached approximately 19000 USD (2). After 2020, under the impact of many external factors, including using high computing power GPU for mining, the exchange rate of bitcoin increased exponentially and reach the ever-highest price of approximately 64000 USD. Although over 7000 cryptocurrencies have emerged (3), bitcoin is still the most popular cryptocurrency among all of them. On 28/05/2022, bitcoin has a total market cap of over 500 billion USD (4) and a transaction volume of over 250000 bitcoins per day (5).

Bitcoin transactions are irreversible, and participants (senders and receivers) are involved only using pseudonyms (addresses). A bitcoin transaction is performed via bitcoin value change between input address and output addresses. In real practice, users often tend to use different addresses for different transactions, and same address will not be used multiple times. This makes bitcoin transactions anonymous and hard to trace.

Due to these special characteristics of cryptocurrency, it provides an ideal channel for illegal parties' money laundering. (3) This has caused government organisations to concern that cryptocurrency, such as bitcoin, may be vastly involved in criminal activities. (6) While some western countries and regions, such as The United States and The European Union, allow transactions using bitcoin. (7) Many countries, such as China and Egypt, have banned all cryptocurrency from making transactions (8). Chinese government even forces

Chinese citizens shut down all bitcoin mining operations. To avoid encouraging illegal services using bitcoin, a tracing software or service is crucial for regulatory authorities and bitcoin service providers. The tracing service can be used in checking legality of bitcoin funds when making payments online or transferring bitcoin to a cryptocurrency exchange.

However, bitcoin provides a public transaction ledger, the blockchain, which means that all transaction histories are visible.(6) The ownership of many addresses along with other information, including transaction records, is now available on many web services, such as WalletExplorer (9). Through backtracking the transaction made by an address, it may be possible to identify the sources of asset present in this address. Sources can be collected as a list of addresses. There are many possible sources of income, including Bitcoin exchange, mining, illegal platforms such as Sheep Marketplace and other owner's address. Other information, such as scam reports regarding the specific address raised, can also be found online. By searching the collected addresses in the report database, any reports related to the address can be found. Illegal activities related bitcoin addresses may be identified via this searching step. Through collecting and analysing online information as well as backtracking, a list of possible sources and the percentage ratio of them in a given bitcoin address can be obtained.

In this project, a robust bitcoin tracing software was designed and implemented in Python. The software design was based on gathering available online information related to a given bitcoin address. Any useful information related was extracted from the internet using web scraping or application programming interfaces (APIs). Since bitcoin blockchain is a public global ledger, all transactions are visible. Information related to the given address is available and can be found on web services such as WalletExplorer (9). In order to trace the source of the bitcoin in the given address, the flow of money was followed. The concept behind the designed tracing method was based a single bitcoin transaction contains one or multiple input and output addresses, as shown in

Figure 1. Through checking the unspent transactions inside the given address (Figure *2*) and traverse the previous transactions quoted in the input section, it is possible to reveal the source of the money. Based on the structure of bitcoin transactions, the author constructed a network of transaction in order to extract the possible source bitcoin addresses. Machine learning was implemented to classify the possible source bitcoin addresses. The extracted information was demonstrated in varies of charts. Finally, the collected source bitcoin addresses were search in the Bitcoin Abuse Database (10) to check if it is linked to any scam information in order to spot out any illegal activity related.

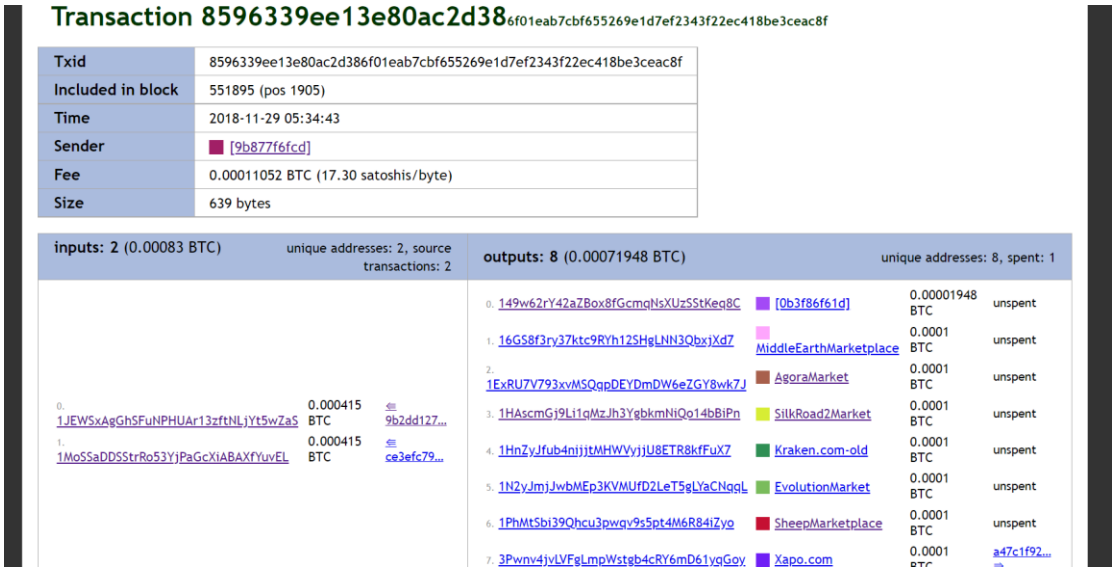

Figure 1: Details of transaction 8596339ee13e80ac2d386f01eab7cbf655269e1d7ef2343f22ec418be3ceac8f (11)

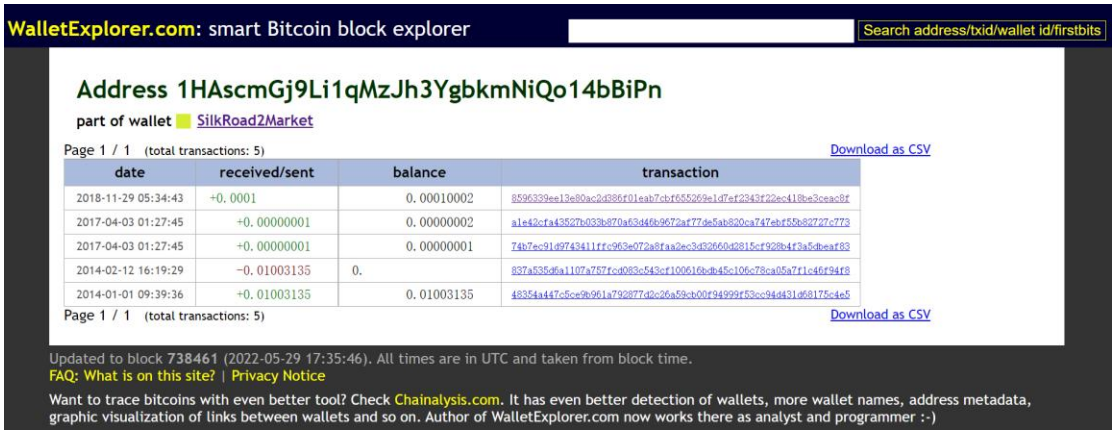

Figure 2: Screenshot of searching a bitcoin address 1HAscmGj9Li1qMzJh3YgbkmNiQo14bBiPn on WalletExplorer. (12)

This thesis is structured as follows: a background literature review related to bitcoin, web scraping and any related work published before was presented in section 2; a detailed software design and how this software was implemented was introduced in section 3; In section 4, the results generated by the software were evaluated with opinions given. Further discussion regarding the software design and implementation were also included; the final conclusion and possible future improvements of this proposed software design were presented in section 5.

2 Background and Related work

2.1 How bitcoin works

In 2008, Satoshi Nakamoto introduced a purely peer-to-peer version of electronic cash which allows online transactions to be made between two parties without going through a financial institution. This cash was later referred as Bitcoin. (1) Bitcoin consists of two cryptographic primitives which are a digital signature scheme and a one-way hash function. The digital signature scheme implements an algorithm called Elliptic Curve Digital Signature Algorithm (ECDSA), and the one-way hash function implements Secure Hashing Function-256 (SHA-256). (6) To generate a digital signature, each user can use his own private keys to generate pairing public keys. A private key, as the name suggested, is unknown to the public, it is used to verify transactions and prove ownership of the users address. Public key, on the other hand, is one of the receiving addresses of a user and can only be verified by the pairing private key. Sometimes, public keys may be referred as addresses or blockchain addresses. A single user can have multiple public and private key pairs, his pseudonyms are the public keys for the signature scheme. Users can use a bitcoin wallet to store bitcoins. However, the wallet itself does not store any bitcoin, it stores all the owner's key pairs to be used in transactions. (13) The balance in a bitcoin wallet is simply the total amount of bitcoins held in all the addresses stored in the bitcoin wallet.

Satoshi describes bitcoin as a chain of digital signatures. To transfer a Bitcoin, the owner needs to sign a hash of previous transaction, the public key of the next owner, and add the information to the end of the chain (Figure 3). (1) However, this does not prevent a Bitcoin from double spending. In order to avoid the problem of double spending, a timestamp server is needed to verify which transaction comes first. Each timestamp is a hash which contains the previous timestamp, this makes all the timestamps form a chain. (1)

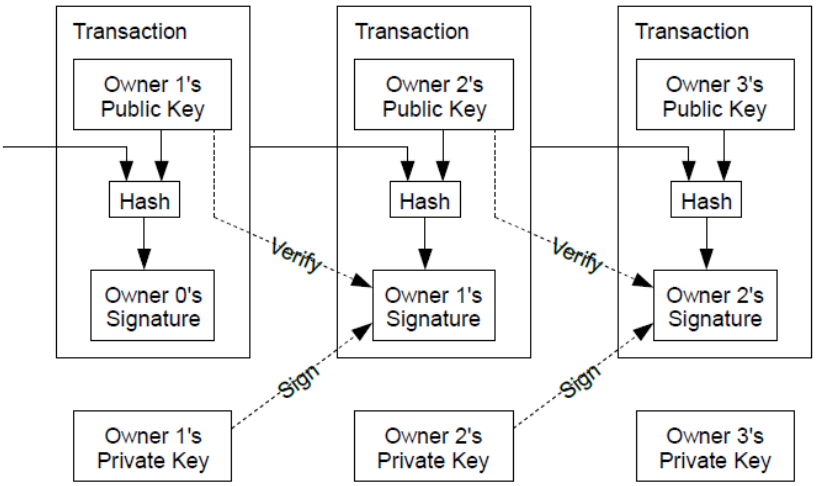

Figure 3: Chain of digital signatures in Bitcoin transaction (1)

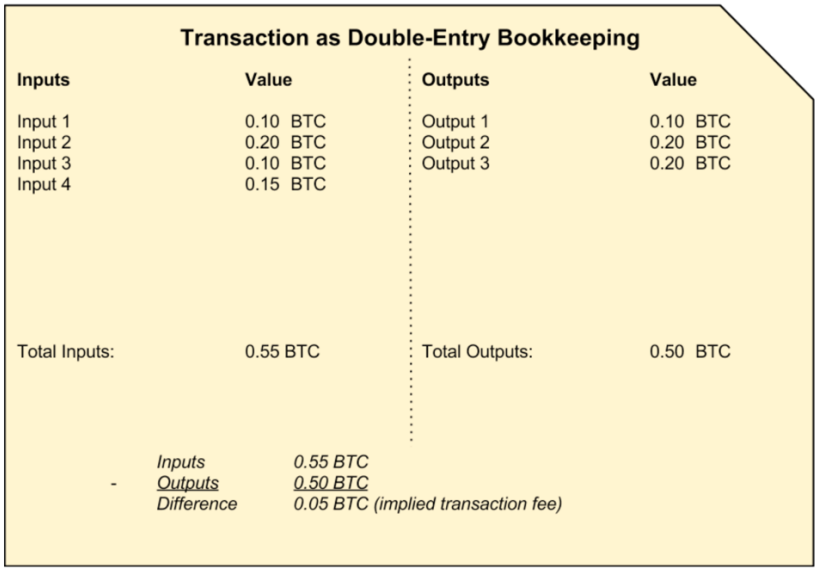

Figure 4: Example of transaction structure as double-entry bookkeeping. (14)

Bitcoin transactions in the bitcoin blockchain are like lines in a double-entry bookkeeping ledger. As shown in Figure 4, each transactions contain one or more inputs in the input section (left) which are debits against the bitcoin address specified in the input section. On the other side, the output section (right) can also contain one or more outputs, which are credits added to the bitcoin address specified in the output section. The total amount of inputs is not required to be equal to the total amount of outputs, output amount can be greater than input amount. The extra output amount implies the transaction fee. The transaction fee will be collected by the miner who includes the transaction

in the bitcoin blockchain. A transaction contains proofs of ownership of bitcoin amount specified in the input section, the ownership is shown in the form of digital signature which anyone in the bitcoin network can validated. Spending any bitcoin is performed through signing a transaction transferring value from a previous transaction to new output addresses.(14) In a typical example transaction data found online (Figure 5), all the inputs have quoted previous transactions signed by the owner.


```
{
  "hash": "b6f6991d03df0e2e04dafffcd6bc418aac66049e2cd74b80f14ac86db1e3f0da",
  "ver": 1,
  "vin_sz": 1,
  "vout_sz": 2,
  "lock_time": "Unavailable",
  "size": 258,
  "relayed_by": "64.179.201.80",
  "block_height": 12200,
  "tx_index": "12563028",
  "inputs": [
    {
      "prev_out": {
        "hash": "a3e2bcc9a5f776112497a32b05f4b9e5b2405ed9",
        "value": "100000000",
        "tx_index": "12554260",
        "n": "2"
      },
      "script": "76a914641ad5051edd97029a003fe9efb29359fcee409d88ac"
    }
  ],
  "out": [
    {
      "value": "98000000",
      "hash": "29d6a3540acfa0a950bef2bfdc75cd51c24390fd",
      "script": "76a914641ad5051edd97029a003fe9efb29359fcee409d88ac"
    },
    {
      "value": "2000000",
      "hash": "17b5038a413f5c5ee288caa64cfab35a0c01914e",
      "script": "76a914641ad5051edd97029a003fe9efb29359fcee409d88ac"
    }
  ]
}
```


Figure 5: Example transaction data from Blockchain.com.(5) Previous transaction id is highlighted.

An important concept in bitcoin transactions is unspent transaction output (UTXO). An unspent transaction output refers to a transaction that can be used as an input in a new transaction (15), which means the bitcoin is unspent and owned by the address specified in the output section. In terms of bitcoin blockchain, there is no actual balance stored for a bitcoin address. There are only scattered UTXOs recorded on different blocks along the whole blockchain. The concept of an address's balance is a derived concept, it is calculated by scanning all the scattered UTXO in the whole blockchain that belong to the specific address.(14) A UTXO can only be spent as whole, which means each

UTXO cannot be divided. The entire amount of bitcoin in a UTXO can only spent all at once in the new transaction, the extract bitcoin (change) can be collected by putting a change address in the output section. In other word, this can be explained as a sender transferring all the money from one of his/her account, the recipient and another account of the sender receiving at the same time.

Satoshi used a proof-of-work system to implement a distributed timestamp server. (1) Transactions are collected into a block, the system requires to hash the transactions in this block as well as previous block information to generate a hash with a certain number of zero bits at the start. A nonce value is incremented until the required hash is generated. The difficulty, which is the number of leading zero bits in the hash, is adjusted by the current hash rate of the bitcoin network. (6) The current hash rate is defined as the current total computational power used of the proof-of-work bitcoin network. (16) Since each block contains information pointing to the previous block, all the blocks forming a chain and is called blockchain. The bitcoin blockchain is a public ledger which keeps track of all transaction history. (13) By dynamically adjusting the difficulty, on average, a new block is created every 10 minutes. (6)

In real practice, in order to spend bitcoins, a user holding multiple addresses (public keys) can use one or more addresses to make any payment. The user creates a transaction containing inputs and outputs (along with other information), the user needs to specify the input addresses which he/she is willing to use in the input section and the receiver's address in the output section. If the funds in the input addresses are not fully used, a new change address needs to be created and included in the output section for returning the changes back to the sender. After the transaction is signed by the user, he then needs to broadcast the message in the bitcoin network. The message will then be confirmed and collected by his peers in the network. His peers will further hash this transaction in the next block in the blockchain.

The creation of a new block in the bitcoin blockchain is called mining. The miners collect all the new transactions into a block. They check the signature validity and ensure no double spending. Each time a miner generates a hash which satisfy the required number of leading zero bits, he/she publishes the resulting new block in the network. His/her peers in the network can vote for its validity by generating the next block based on this block. A majority decision is represented by the longest chain in the network which is the greatest proof-of-work effort. (1)

The miners can receive a processing fee (transaction fee) by taking the change left in each transaction when a change address is not specified. However, the transaction fee is not mandatory in a transaction. The main motivation of miners is the incentive of the bitcoin network and is the reason why the process is called mining. The miner receives a bonus of bitcoin when creating a new block on the blockchain since he/she has done some intensive one-way computational work. (6) This is more or less similar to mining gold, workers make efforts in digging and receive gold as a return. This provides a way to distribute bitcoins into the circulation in the network, since there is no central authority to issue bitcoins. (1) The miners receive the reward by adding a special transaction into the new block he/she generates, called coinbase transaction or generation transaction. This special transaction normally appears as the first transaction in the block. Unlike regular transactions, a coinbase transaction does not consume an UTXO, which means no UTXO is quoted in the input section. Instead, only one input called coinbase is included. The bitcoin generated is paid to the miner's address which is specified as output in the coinbase transaction. (14)

The bitcoins issued as rewards is dependent on the length of the bitcoin blockchain. The reward is halved each time after 210000 more blocks created. The first ever block in the bitcoin blockchain was created by the inventor Satoshi Nakamoto, and the original reward was 50 bitcoins per block. (17) Until June 2022, the reward of generating a new block is 6.25 bitcoins. (18) The total

number of bitcoins ever be issued will be 21 million, the estimate time of mining all the available bitcoins is the year of 2140. (19)

2.2 Web scraping

At present, the internet contains enormous amount of information including texts, graphical and videos. The web pages displaying this information can only be viewed using a web browser. However, the web browser doesn't provide functionality such that the user can save data for personal use. The only option for the user is to manually browse the web page and save the data which is a cumbersome task to do. (20)

Instead of manually copying data, an automatic process can be used, it is called web scraping (also known as screen scraping, web data extraction and web harvesting etc.) (20) Web scraping consists of three main stages: website analysis, website crawling, and organisation. As shown in Figure 6, several web technologies and at least one popular programming language (such as Python) are required to perform web scraping. Even though web scraping is an automatic process, it is not fully automatic, the three phases stated still need some extent of human supervision. (21)

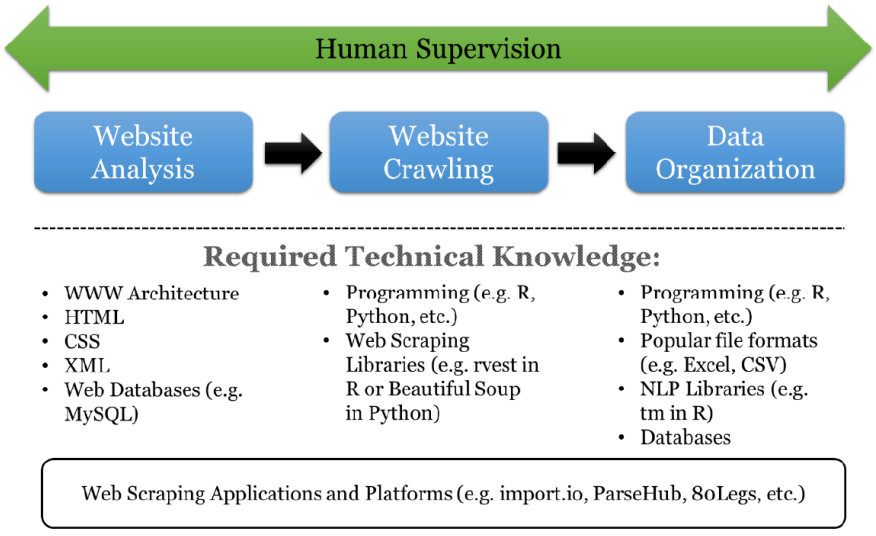

Figure 6: Web scraping (21)

Due to the difference in websites, website analysis needs to be performed first to understand how data is stored and presented on a specific website. This phase requires knowledge of the World Wide Web architecture, mark-up languages (HTML, CSS, XML, etc.), and query languages (such as MySQL). After knowing how the data is stored and presented, an automatic running programme can be deployed to browse website and retrieve data. This phase is called web crawling. The programme can be developed in multiple languages based on the preferences of developer, popular examples include R and Python. Useful open-source libraries are 'rvest' package in R and Beautiful Soup library in Python. After data has been retrieved from selected websites, it needs to be processed and stored for future analysis. This process is called data organisation. Ways of storing data can be XSLX or CSV files, also data can even be stored in a database. (21)

In performing web scraping, the legality and ethics of retrieving data must not be overlooked. Failure in paying attention to these aspects may lead to legal controversies and lawsuits. (21) Krotov et al. pointed that the legality of web scraping is still a grey area in the legal field, the practice of web scraping is guided by set of related legal theories and laws, such as 'copyright infringement', 'breach of contract', etc. As for ethics, possible harmful consequences of web scraping can be breaching individual privacy, diminishing value for the organisation, etc. (22)

2.3 Related Work

Based on the visibility of the whole bitcoin blockchain and other real-world information, many researchers are trying to break the anonymity of bitcoin by clustering different addresses and identify the ownership of different address groups. (23,24) Lin et al.(23) utilised machine learning to predict the ownership of bitcoin addresses. Dataset was collected and divided up into several entity categories. Different machine learning models was trained using their dataset and compared. Among all the machine learning models they trained, artificial

neural network model constructed using Keras (25) demonstrated the best performance (F1-score) in entity-based scheme. Since this project aims to classify into different entity classes for any bitcoin address collected, artificial neural network was chosen to be the classifier based on their research result.

3 Software Design and Implementation

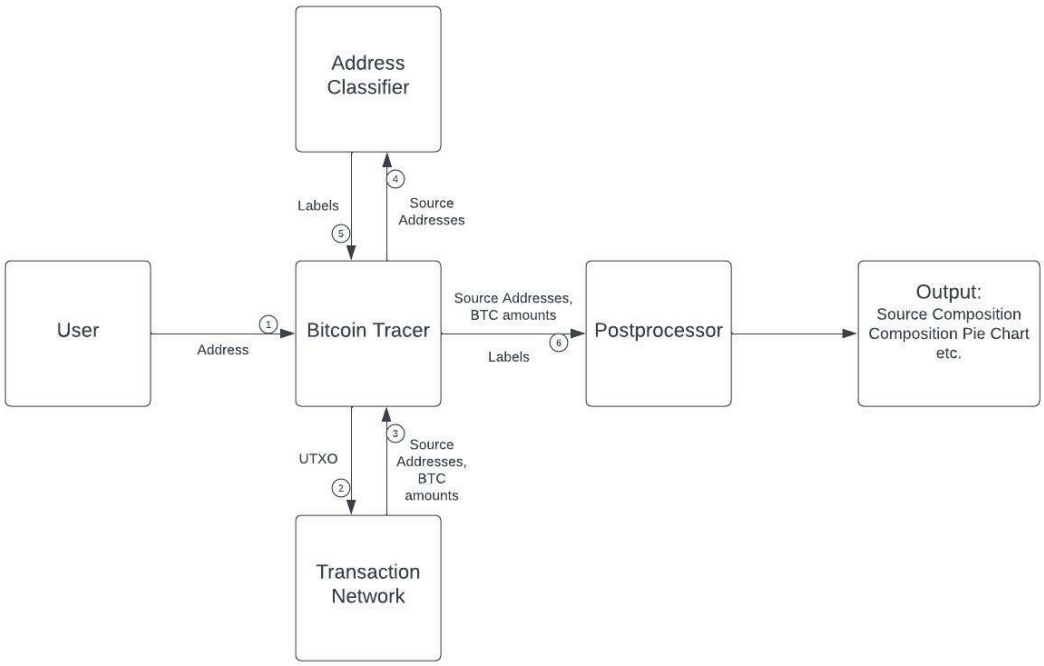

Figure 7: Bitcoin tracer architecture

The architecture of the proposed bitcoin tracer design is shown in Figure 7. The order of tracing process was labelled as shown. Firstly, as the user input a bitcoin address into the bitcoin tracer, the tracer checks the legality and balance of the provided address. Any illegal or unused address will be alarmed and rejected. Also, since the process is based on tracing back of any existing fund in the bitcoin address provided, any address with zero balance, i.e. with no unspent transaction output (UTXO), will be rejected. After the bitcoin tracer receives a legal bitcoin address, it will construct a transaction network base on the UTXO of the provided address. After the transaction network is constructed, all the source addresses are collected and returned back to bitcoin tracer. The source addresses are then passed to a pretrained classifier to predict the nature of all the source addresses. The address classifier is a pretrained artificial neural network model which gives a label among 5 different categories (exchange, mining pool, gambling, darknet markets and services/others) to a certain bitcoin address. After the classification process is finished, all the

collected data and labels are passed to postprocessor, where all the collected results are summarised and presented, including source composition table, composition pie chart, etc.

Too further explain the design of bitcoin tracing concept this project has introduced, the following subsections will explain more in details regarding the design and implementation about two major parts of the concept (transaction network and bitcoin address classifier). Finally, in the last subsection, details about how the collected information is processed will be explained.

3.1 Transaction Network

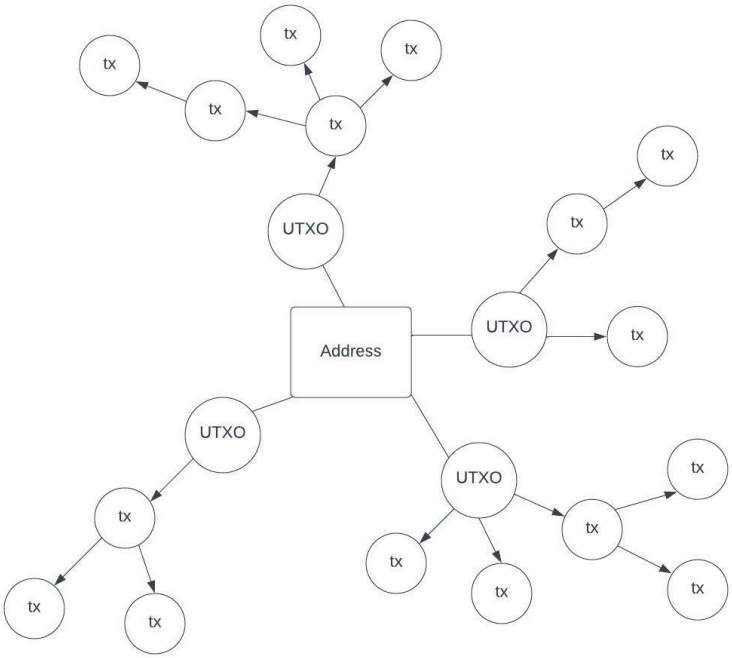

Figure 8: Schematic diagram of a transaction network of a given bitcoin address, trace depth = 2.

In order to trace the flow of bitcoins, the bitcoins from the previous transactions quoted in the input section of a given transaction is traced back first. Since any given address with a positive balance has a certain number of UTXOs (at least one), each UTXO can act as an individual origin to trace back all the previous transactions for this particular UTXO, the summation of every sub-network

constructed by each UTXO forms a transaction network of the given bitcoin address. A schematic diagram of this network is illustrated in Figure 8.

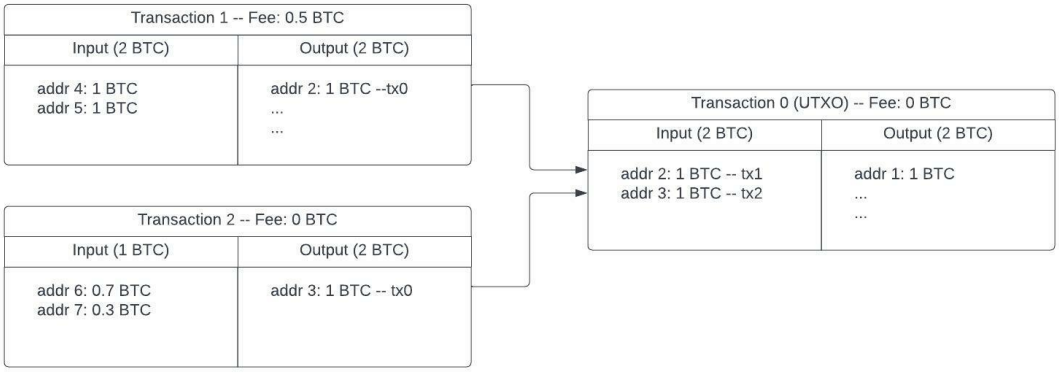

Figure 9: Simple example of a transaction sub-network. *Addr 1*: tracing bitcoin address

Based this network, varies of information can be collected, such as source addresses and their corresponding bitcoin amounts. In this design, the source addresses are defined to be all the input addresses in leaf transactions in the transaction network. To illustrate this in more detail, a simple example was constructed in Figure 9. Assume *addr 1* is the bitcoin address which needs to be traced, *transaction 0* is one of the UTXOs and the depth of the sub-network is only 1. The source addresses of the 1 BTC fund inside *addr 1*'s balance is therefore defined to be: [*addr 4*, *addr 5*, *addr 6*, *addr 7*].

To calculate the corresponding bitcoin amount, the contribution of each source address needs to be calculated accordingly. The contribution factor of each address inside a transaction can be defined as:

$$c_i = \frac{a_i}{\sum_{i=1}^{n} a_i}$$

Where *c* is the contribution of address *i*, *a* is the input amount made by address *i*, *n* is the number of input addresses in the transaction. By multiplying all the previous contribution and the UTXO amount, the final corresponding bitcoin amount of a source address can be calculated:

$$C = b \prod_{j=0}^{s} c_j$$

Where, *C* is the corresponding bitcoin amount, *b* is the UTXO bitcoin amount, *c* the contribution factor, *s* is the depth of leaf transaction in the transaction network. So, the tracing result for the demonstrated example in Figure 9 can be calculated, the calculation is shown in Table 1. As demonstrated in the table, the sum of all the bitcoin amounts is equal to the UTXO amount of *addr 1*, which is 1 BTC. Thus, it can be assumed that, for the 1 BTC fund in *addr 1*, 0.25 BTC is from *addr 4*, 0.25 BTC is from *addr 5* and so on.

Source Address	Bitcoin Amount
addr 4	$1 \cdot \frac{1}{2} \cdot \frac{1}{2} = 0.25$
addr 5	$1 \cdot \frac{1}{2} \cdot \frac{1}{2} = 0.25$
addr 6	$1 \cdot \frac{1}{2} \cdot \frac{7}{10} = 0.75$
addr 7	$1 \cdot \frac{1}{2} \cdot \frac{3}{10} = 0.15$
Sum	1

Table 1: Calculation of corresponding bitcoin amount of example shown in Figure 9

One thing to be mentioned is that the result of the trace varies as the structure of the transaction network varies. The structure of a transaction network can be affected by multiple factors, such as tracing depth and early stropping mechanisms, in construction of the network. In the designed network, tracing depth is defined as the maximum depth of the construction of transaction network. Since almost all the transactions have at least one referenced previous transaction output (except the case that it is a coinbase transaction where the input is empty), the maximum depth of the network must be introduced to avoid constructing a transaction network that is too large. Based on this restriction, the network can be constructed with a confined size. However, sometimes the tracing depth may be set too large to collect any meaningful information, which means the transaction network has redundant transactions. Thus, early stopping mechanisms are also needed in construction of transaction network to remove any redundant transaction nodes.

The early stopping mechanisms implemented in this project include stopping when tracing to a coinbase transaction and stopping with smallest bitcoin unit (amount). As mentioned above, in a coinbase transaction, there is no previous transactions. The bitcoin transferred in a coinbase transaction is considered to be mined, which is defined as a reward for the miner's computational power. Whenever a coinbase transaction is reached, the network will not construct any further in this particular branch since the tracing has reached the true source of the bitcoin. Also, during construction of transaction network, as the trace progress deeper, the corresponding contribution bitcoin amount of each source address becomes smaller. This is due to the multiplication of contribution factor in each level of depth. In the case that the trace depth is too big, the corresponding bitcoin amount may be smaller than 10^{-8} BTC. Since the smallest unit of bitcoin is 1 satoshi which is defined as 10^{-8} BTC, it is meaningless to trace bitcoin with amount any further smaller than 1 satoshi. So, in the implementation of transaction network, the construction of the branch will stop when an address's contribution is smaller than 1 satoshi. Meanwhile, if the tracing address has a large balance, for example over 1000 BTC, the transaction network can be too dense since the network is constructing with a percision of 1 satoshi. Due to the fact that the address has a large balance, it may be unnecessary to trace any fund that is only a few satoshi. So, in order to get a greater picture of the network, a parameter called *minimum_amount* was introduced in the implementation. By stopping the construction of a branch when the address's contribution is smaller than *minimum_amount*, it is able to remove any unnecessary transaction nodes in the network, the tracing process can be tailored to the user's need.

The construction of transaction network introduced in this thesis was implemented using recursion. API requests were sent to Blockchain.com to obtain the UTXOs of the tracing address first. From the UTXOs of the tracing address, the network will grow by recursively constructing new nodes base on each previous transactions quoted in the input section. The base cases applied

were implemented as discussed before: when the network reached the maximum depth of trace; when the contribution of an address was less than *minimum_amount*; when the network reached a coinbase transaction. Transaction information were also acquired via API requests provided by Walletexplorer.com (12), API requests were sent with an interval of 0.5s to match the request rate limit.

3.2 Bitcoin Address Classifier

Since the bitcoin blockchain is a public ledger, all the transaction history is available on the bitcoin blockchain. The transaction history of any given bitcoin address can be obtained from webservices such as Blockchain.com (5). By extracting statistic features from the transaction history, it is possible to train an artificial neural network model with known addresses.

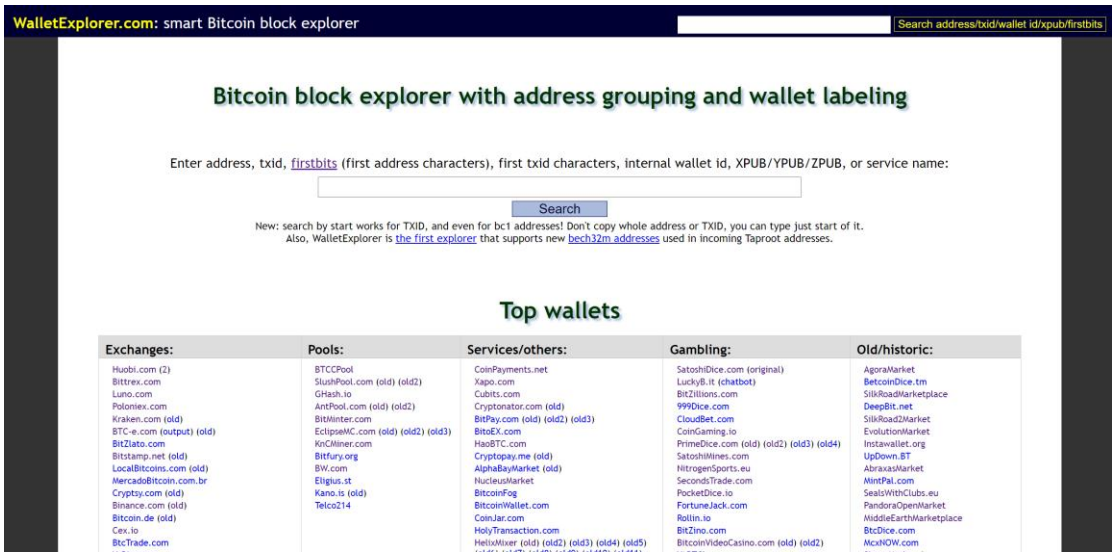

Figure 10: Main page of WalletExplorer (12)

Since there was no available dataset available on the internet, to implement address classifier in this project, data collection was performed first. As shown in Figure 10, WalletExplorer (12) provides number of lists of address belonging to different entities. By crawling through different lists of addresses, a total number of 9976 addresses with known entities and classes were collected. 5 different classes were selected including exchanges, mining pools, services or

others, gambling, and darknet markets. Details about the collected address dataset is shown in Table 2. The specific entities selected for different classes were shown in Table 5 in the Appendices section.

Class	# of Entities	# of Addresses
Exchanges	10	2000
Mining Pools	10	2000
Services/others	10	2000
Gambling	10	2000
Darknet Market	9	1976
Total	49	9976

Table 2: Details of address dataset

After collecting the entity dataset, features of the transaction for each address were computed by requesting full transaction history from Blockchain.com (5). Each API request sent to Blockchain.com was set to have a 10s interval to match the request rate limit of the website. 9 types of features were selected to summarise the transaction history of an address, as shown in Table 3. In total there was 38 features extracted, the full list of features is given in Table 6 in Appendices section.

Feature	Description
$\mathrm{N}_{\mathrm{transactions}}$	Number of transactions
$lifetime$	Block height range among all transactions
fee	Transaction fee
$size$	Data size of each transaction
$weight$	Transaction size in weight units
$input\ size$	Number of input addresses in each transaction
$output\ size$	Number of output addresses in each transaction
$input\ amount$	Bitcoin amount in input section in each transaction
$output\ amount$	Bitcoin amount in output section in each transaction

Table 3: Feature types selected to summarise a bitcoin address's transaction history

After the whole dataset was collected, the dataset was pre-processed using the Python machine learning library – Scikitlearn(26). The data was split into

training and testing datasets using *train_test_split()* with fraction of 0.9/0.1 (train/test). The label (class) data in both datasets was fit and transformed to categorical data using *LabelEncoder()*. The feature data in training dataset was fit and transformed using *StandardScaler()*, the feature data was only transformed using the fitted scaler.

The artificial neural network model was implemented using another Python machine learning library – Keras(25). The construction of neural network started with a single hidden layer with 100 neurons, the structure of the neural network was tuned based on the accuracy of the constructed model. The number of neurons in each hidden layer and number of hidden layers were gradually increased to improve accuracy. As the complicity of the network increased, the accuracy of the model reached a plateau, further increase in complicity led to increase in the level of overfitting. The final neural network consists of 3 fully connected hidden layers with 300 neurons in each layer. The input layer size was set to 38 to match with the data dimension, the output layer size was set to 5 to match the number of predicting classes. '*Softmax'* was chosen as the activation function of the output layer. The model was compiled with '*categotical_crossentropy*' as the loss function and '*adam*' as the optimiser.

3.3 Result Processing

After a list of source addresses were extracted, the labelled addresses and the corresponding bitcoin amount were passes to the postprocessor. The data was summarised according to each specific category, total amount of bitcoin for each category was calculated. Pie charts were generated based on the total amounts. Also, the tracing depth was varied for different tracing addresses. A line chart can be plotted to see the composition change as the tracing depth. Finally, all the collected source addresses were checked against Bitcoin Abuse Database (10), to see if there is any illegal activity report raised.

4 Evaluation

Through application of the proposed design, useful information can be extracted from the results. Based on this extracted information, it was possible to trace and analyse bitcoins present in any bitcoin addresses. The process time was around 10-120 minutes, based on different transaction network structures. The process time was acceptable. The process time was dominated by the classification process. This was because the classification process requires extracting features of its transaction history, each extraction takes a fixed 10s due to the API request limit of the online blockchain data provider.

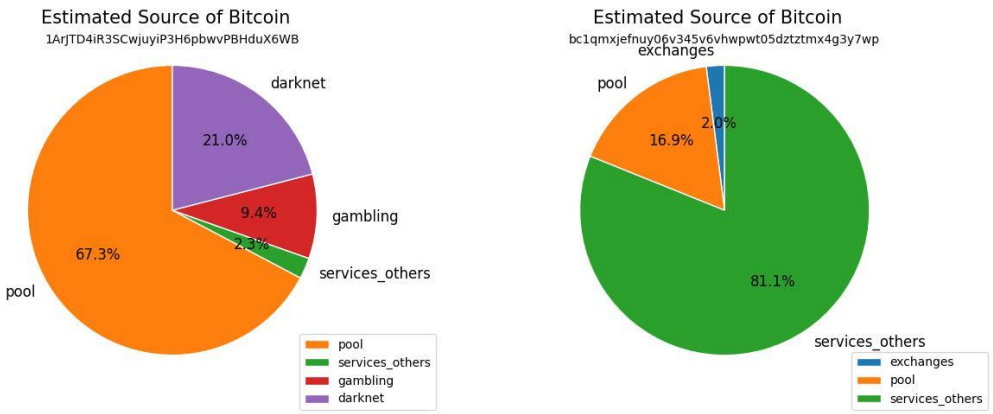

Figure 11: Examples of tracing result. Left: 1ArJTD4iR3SCwjuyiP3H6pbwvPBHduX6WB, *tracing depth* = 10, *minimum amount* = 0.005 BTC; Right: bc1qmxjefnuy06v345v6vhwpwt05dztztmx4g3y7wp, *tracing depth* = 5, *minimum amount* = 100 BTC.

Figure 11 illustrates some example results produced by the bitcoin tracer. By summing up the contribution for all the addresses presented in each category, pie charts were generated. From the pie chart, it is clear to visualise the composition of funds present in the tracing address at a certain tracing depth. By varying the depth of trace, the user is able to peel off the composition structure of the fund present in the tracing address. This analysation process can be further summarised into a single line chart to see the change in composition as the depth changes. In Figure 12, the tracing address (*1ArJTD4iR3SCwjuyiP3H6pbwvPBHduX6WB*) is an address own by

Bitpay.com according to Walletexplorer.com(12), the analysation process was performed with a *minimum amount* = 0.005 BTC. As the figure illustrated, the result shows that almost all of its fund (over 90%) was from darknet markets. As the depth of trace went deeper, it was shown that around 35% percent of the fund was from services or others categories before the fund flowed into this tracing address. However, this portion of fund was further proved to be also from darknet markets. Finally, as the tracing progressed very deep, it shows that the fund coming from darknet market was originated from mining pools. This finding was reasonable because all the bitcoins are issued by rewards paid to the miner, tracing back of any bitcoin transactions will all eventually reach a coinbase transaction or a transaction made by mining pool. After the tracing went into mining pools, the composition stopped having any variation. This was because mining pools often manage larger number of addresses to issue accounts for miners or transfer mining fees to miners, without the internal data base, the tracer will be lost in the large number of addresses owned by mining pools.

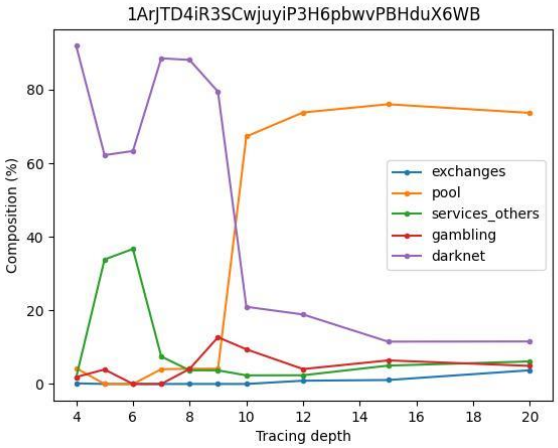

Figure 12: Composition of different classes for 1ArJTD4iR3SCwjuyiP3H6pbwvPBHduX6WB. *Minimum amount* = 0.005 BTC

Apart from tracing a single standalone bitcoin address, this designed software can also be used in a scaled way. Since a typical bitcoin user may have more than one address stored in a bitcoin wallet, it is possible to identify the wallet according to the provided bitcoin addresses. After identifying the specific wallet,

all the addresses in this wallet can be collected. For each address that have a positive balance, a transaction network can be constructed for tracing. With this, a more detailed and advanced knowledge can be obtained for the given address as well as the bitcoin wallet. This function was not implemented in this project due to the process speed of the software. However, the author believed that the proposed design is capable of doing this. To implement this function, only a simple step is needed to be added in. After the user provides the tracing bitcoin address, the wallet which the address belongs is checked. Apart from tracing the given bitcoin address, other addresses inside this wallet will also be traced. In the future, if the data accessing speed can be improved, it is believed that this design can perform to its full potential.

4.1 Transaction Network

Overall, the design of this transaction network was robust. The design allows the user to tune the structure of the network using *tracing depth* and *minimum amount* as parameter. By changing these parameters, the user is able to alter the precision of the tracing process and focus on the specific information which they are interested in. In the original design, the construction of the network was different. One more stopping mechanism was included in the original design, the address classifier was used in construction of the network. To illustrate this design, an example is shown in Figure 13 (*tracing depth* = 4). With the current design, the tracer will collect all the input addresses in the right most end of the top and bottom branch. However, the transaction coloured in red may contain an illegal address in the input section. If the transaction network is not processed with *tracing depth* = 3, the current design will miss out the illegal source. If the interest of the trace is to identify any illegal source of fund, the network should stop constructing whenever an address own by illegal services occurs as an input, since any money coming out of an illegal source is ‘contaminated’ despite previous any sources.

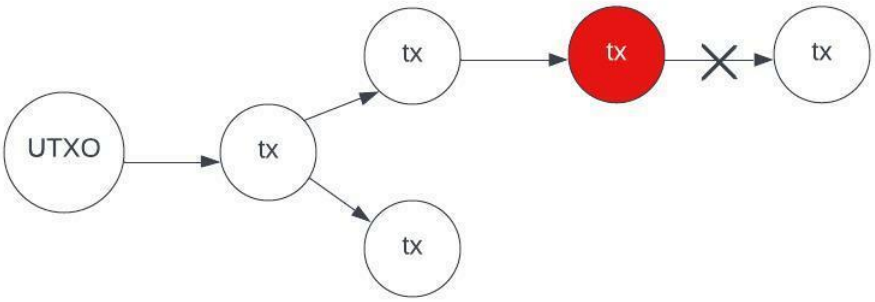

Figure 13: Schematic diagram of bitcoin tracing involving illegal address. Transaction containing illegal addresses is coloured in red. *tracing depth* = 4

To implement this, the original design is used to classify all the input addresses during the construction of network. Whenever an illegal address was classified, the construction will stop in the branch. For the example in Figure 13, the construction on the top branch will stop at the transaction coloured in red, even when the *tracing depth* is set to be greater than 3.

However, this original design was not implemented due to the following reasons. During the time of this project, all the blockchain data must be access through webservices. Most webservices have request limits for their API, the fastest and easiest API available to use was provided by Blockchain.com (5). Their request limit was found to be 10s per request. To implement the design mentioned above, all the input addresses in each transaction nodes inside the network must be classified. A normal small network may contain hundreds of transaction node, which means in total thousands of addresses must be classified. Even though some addresses may be repeating, they are just a very small portion among the whole transaction network. Therefore, the estimated overall process time will be over 2 hours if the classifier is included in the network construction process. The overall process time is too long and is not acceptable for this software and project.

What's more, even if the design of the network has tried to cover all the possible circumstances. The accuracy this network design is inevitably affected by the

mixing services. Mixing services combine larger number of input and outputs together to retain the anonymity of bitcoin, which makes the tracing process hard and may mislead the tracer. There is no easy way to overcome this. A possible way to improve the design is also to add an address classifier during transaction network construction (as introduced before). By stop the construction when a mixing service address is found, the network stop constructing even further. Another possible way to improve the design is to introduce a method to identify a mixing transaction, for example, implement the method which Wu et al.(27) proposed.

4.2 Bitcoin Address Classifier

When evaluating the bitcoin address classifier, the performance of the neural network model was good. The accuracy of the model against held-out testing dataset was 86%. In Table 4, metric of the neural network's performance against the held-out testing dataset is shown.

	precision	recall	f1-score	support
Darknet markets	0.79	0.8	0.79	200
Exchanges	0.92	0.88	0.9	202
Gambling sites	0.79	0.79	0.79	186
Mining pool	0.91	0.97	0.94	196
Services/others	0.87	0.86	0.86	214
Accuracy			0.86	998
Macro average	0.86	0.86	0.86	998
Weighted average	0.86	0.86	0.86	998

Table 4: Metrics of the neural network model's performance against test dataset

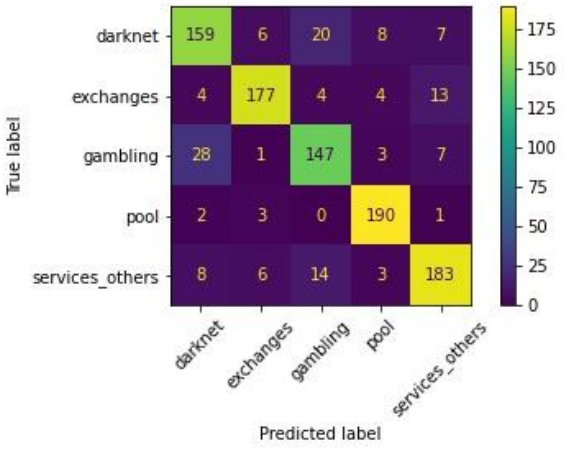

Figure 14: Confusion matrix of the neural network model's prediction against test dataset

As the metrics (Table 4) and confusion matrix (Figure 14) demonstrated, the model performed better in classifying exchanges, mining pools and services/others. This model was especially better at classifying mining pools, which had a high F1-score of 94%. However, the aim of this project focused more on identifying illegal darknet and gambling addresses. The performance of the trained model, evaluated based on the F1-scores, was slightly lower in classifying darknet markets and gambling sites. The confusion matrix showed that the model was confused when classifying between darknet markets and gambling sites. This may be caused by all illegal services having similarities. For example, illegal services are more likely to use their addresses with shorter period of time or only transfer small number of bitcoins in a transaction. What's more, gambling sites and darknet markets may even have shared addresses, which may also cause confusion between the two classes.

Even though the model had a high accuracy against testing dataset, the model may still not be ideal for bitcoin address classification. The features extracted from the transaction history were only based on basic statistics, more advanced features such as transaction moments proposes by Lin et al.(23) can be added for a better summarisation of transaction history characteristics.

The 5 classes selected in this project was limited by the availability of data on the internet. WalletExplorer(12) only provided address with 5 classes. However, bitcoin addresses can be classified into more classes, such as faucet, mixing service, etc. Through introducing more classes in the model, it may be able to improve its performance. Also, as discussed before, the model was confused between darknet markets and gambling site. It may be beneficial to combine the two categories into one, which can be named as ‘illegal services’. Furthermore, if the purpose of use is only identifying the funds from illegal services, instead of classifying multiple classes, the neural network model can be switched to binary classification model to predict if the address is from illegal services.

Regarding to the training dataset collected, there are two ways which it can be further improved. The labelled addresses provided by WalletExplorer(12) were only updated until 2016 according to its information page. At the time of this project, which was 2022, many services listed such as some mining pools and darknet markets were shut down. Existing services may have different way of operation, this may alter the transaction history characteristics of their addresses. So, the data used in this project can be slightly out of date and can be improved by collecting a set of more recent addresses used.

Due to the data collection process requires sending API requests to the data providing websites, such as Blockchain.com, the speed of data collection was limited by the request rate. The request rate was set with at least 10s interval to avoid being banned by the server. Thus, with this request speed, data for only 9976 addresses were extracted. The dataset can be further enlarged to get a more reliable dataset. In the future, the data collection process can be largely improved via higher frequency API provided by webservice. It will be even more beneficial, if the whole blockchain data is available on a local server.

5 Conclusions

In summary, a software was designed and implemented to trace and analyse a given bitcoin address. It was demonstrated that this software is capable of tracing funds present in a given address, and also collecting source address using a transaction network. A neural network model was trained and implemented in the software to classify the collected source addresses. With the classification results, it was demonstrated that the software was able to analyse the fund composition variation with change of tracing depth. The whole design is scalable and robust. By simply repeat the tracing process, the software is able to trace and analyse all the addresses inside a wallet when only one of its addresses is provided. The design allows the user to tune the tracing structure and process with various parameters. Even though some extra factors were not implemented, the design allows multiple extra tunning mechanisms to be added.

In the future, the design can be improved on different parts. The transaction network can be improved by adding a classifier into the construction process. The construction process can be set, such that the construction process of the branch will terminate when a certain class of address appears. Additional model can be implemented to identify a mixing service transaction to further improve the accuracy of the tracing process. What's more, the artificial neural network model can be further improved by training the model with more advanced statistical features, as well as a more up to date and larger dataset. For both the runtime speed of the software and construction of a machine learning classifier, it will be very beneficial if there is a possible method of faster accessing bitcoin blockchain data online or even on a local server. This project is a preliminary study of building a software to trace any bitcoin, even though the software still has many factors that can be improved, it is certain that the design has a large potential in tracing bitcoin flow. The author believed that the designed software can be developed even further.

6 Bibliography

1. Nakamoto S. Bitcoin: A Peer-to-Peer Electronic Cash System. SSRN Electron J. 2008;
2. Bitcoin Price Chart (BTC) | Coinbase [Internet]. @coinbase. 2022. Available from: https://www.coinbase.com/price/bitcoin
3. Liu XF, Jiang XJ, Liu SH, Tse CK. Knowledge Discovery in Cryptocurrency Transactions: A Survey. IEEE Access. 2021;9(2):37229–54.
4. Coinmarketcap. Bitcoin [Internet]. CoinMarketCap. 2022. Available from: https://coinmarketcap.com/currencies/bitcoin/
5. Blockchain Explorer [Internet]. www.blockchain.com. 2022. Available from: https://www.blockchain.com/explorer?view=btc
6. Meiklejohn S, Pomarole M, Jordan G, Levchenko K, McCoy D, Voelker GM, et al. A fistful of Bitcoins. Commun ACM. 2016;59(4):86–93.
7. Bajpai P. Countries Where Bitcoin Is Legal & Illegal (DISH, OTSK) [Internet]. Investopedia. 2021. Available from: https://www.investopedia.com/articles/forex/041515/countries-where-bitcoin-legal-illegal.asp
8. Orji C. The countries where Bitcoin and crypto are banned or restricted [Internet]. euronews. 2022. Available from: https://www.euronews.com/next/2022/04/27/bitcoin-ban-these-are-the-countries-where-crypto-is-restricted-or-illegal2
9. WalletExplorer.com: smart Bitcoin block explorer [Internet]. Walletexplorer.com. 2022. Available from: https://www.walletexplorer.com/
10. Bitcoin Abuse Database [Internet]. www.bitcoinabuse.com. 2022. Available from: https://www.bitcoinabuse.com/
11. WalletExplorer.com [Internet]. www.walletexplorer.com. 2022. Available

from:
https://www.walletexplorer.com/txid/8596339ee13e80ac2d386f01eab7cbf655269e1d7ef2343f22ec418be3ceac8f

12. WalletExplorer.com [Internet]. www.walletexplorer.com. 2022. Available from:
https://www.walletexplorer.com/address/1HAscmGj9Li1qMzJh3YgbkmNiQo14bBiPn

13. Manimuthu A, Raja Sreedharan V, Rejikumar G, Marwaha D. A Literature Review on Bitcoin: Transformation of Crypto Currency into a Global Phenomenon. IEEE Eng Manag Rev. 2019;47(1):28–35.

14. Antonopoulos AM, Media O. Mastering bitcoin : programming the open blockchain. O'reilly PP - Beijing ; Boston ; Farnham ; Sebastopol ; Tokyo; 2018.

15. Unspent Transaction Output (UTXO) [Internet]. Binance Academy. 2022. Available from:
https://academy.binance.com/en/glossary/unspent-transaction-output-utxo

16. Fantazzini D, Kolodin N. Does the Hashrate Affect the Bitcoin Price? J Risk Financ Manag. 2020;13(11):263.

17. Kroll J a, Davey IC, Felten EW. The Economics of Bitcoin Mining, or Bitcoin in the Presence of Adversaries. Twelfth Work Econ Inf Secur (WEIS 2013). 2013;(Weis):1–21.

18. Block Reward [Internet]. Investopedia. 2022. Available from:
https://www.investopedia.com/terms/b/block-reward.asp

19. Meynkhard A. Fair market value of bitcoin: Halving effect. Invest Manag Financ Innov. 2019;16(4):72–85.

20. Singrodia V, Mitra A, Paul S. A Review on Web Scrapping and its Applications. 2019 Int Conf Comput Commun Informatics, ICCCI 2019. 2019;

21. Krotov V, Johnson L, Silva L. Tutorial: Legality and ethics of web

scraping. Commun Assoc Inf Syst. 2020;47(1):539–63.

22. Krotov V, Silva L. Legality and ethics of web scraping. Am Conf Inf Syst 2018 Digit Disruption, AMCIS 2018. 2018;(May).

23. Lin YJ, Wu PW, Hsu CH, Tu IP, Liao SW. An Evaluation of Bitcoin Address Classification based on Transaction History Summarization. ICBC 2019 - IEEE Int Conf Blockchain Cryptocurrency. 2019;302–10.

24. Ermilov D, Panov M, Yanovich Y. Automatic bitcoin address clustering. Proc - 16th IEEE Int Conf Mach Learn Appl ICMLA 2017. 2017;2017-Decem:461–6.

25. Chollet F, others. Keras. 2015.

26. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine Learning in Python. J Mach Learn Res. 2011;12:2825–30.

27. Wu L, Hu Y, Zhou Y, Wang H, Luo X, Wang Z, et al. Towards understanding and demystifying bitcoin mixing services. Web Conf 2021 - Proc World Wide Web Conf WWW 2021. 2021;33–44.

7 Appendices

Class	Exchanges	Pools	Services/others	Gambling	Darknet Market
Entity	Huobi.com	BTCCPool	CoinPayments.net	SatoshiDice.com	AgoraMarket
	Huobi.com-2	SlushPool.com	Xapo.com	SatoshiDice.com-original	SilkRoadMarketplace
	Bittrex.com	SlushPool.com-old	Cryptonator.com	BitcoinVideoCasino.com	SilkRoad2Market
	Luno.com	SlushPool.com-old2	Cubits.com	NitrogenSports.eu	SheepMarketplace*
	Poloniex.com	AntPool.com	HaoBTC.com	CoinGaming.io	MiddleEarthMarketplace
	Kraken.com	AntPool.com-old2*	HelixMixer	SatoshiMines.com	CannabisRoadMarket
	Binance.com	KnCMiner.com	BTCJam.com	PrimeDice.com	PandoraOpenMarket
	MercadoBitcoin.com.br	BitMinter.com	GreenRoadMarket	Betcoin.ag	AbraxasMarket
	Bitstamp.net	GHash.io	NucleusMarket	CoinRoyale.com	BlueSkyMarketplace
	Cex.io	BW.com	HelixMixer-old	SwCPoker.eu	
	AntPool.com-old2	SheepMarketplace	other		
# of Addresses	176	400	200		

Table 5: Entity list and number of number addresses for address dataset in training artificial neural network model

Feature					
N_transactions					
lifetime					
fee_mean	fee_median	fee_std	fee_min	fee_max	fee_max-min
size_mean	size_median	size_std	size_min	size_max	
weight_mean	weight_median	weight_std	weight_min	weight_max	
vin_sz_mean	vin_sz_median	vin_sz_std	vin_sz_min	vin_sz_max	
vout_sz_mean	vout_sz_median	vout_sz_std	vout_sz_min	vout_sz_max	
inputsAmount_mean	inputsAmount_median	inputsAmount_std	inputsAmount_min	inputsAmount_max	
outputsAmount_mean	outputsAmount_median	outputsAmount_std	outputsAmount_min	outputsAmount_max	

Table 6: List of features extracted from a bitcoin address's transaction history

8 User guide

- ✓ To use the software, simply clone the project files:

  git clone https://gitlab.doc.ic.ac.uk/zh1516/bitcoin_trace.git

  cd bitcoin_trace

- ✓ If the Python system doesn't have pip installed, do:

  python -m ensurepip –upgrade

  or,

  python get-pip.py

- ✓ Then install all the packages according to the configuration file requirement.txt:

  pip install -r requirements.txt

- ✓ Generate your own token in BitcoinAbuse.com and make a Python file named *my_token.py* in *bitcoin_trace* directory.

  my_token.py:

  bitcoinAbuseToken = {your_token_here}

- ✓ Before running the programme, make sure the parameters in main.py is what you want. To run the programme:

  python main.py

- Alternatively, you can open *bitcoin_trace* as project directory in IDEs such as PyCharm and continue work from there.