# Bitcoin Network Analysis

**Mihir Sutariya**
Indian Institute of technology Gandhinagar, Roll No: 20110208

**Sahil Agrawal**
Indian Institute of technology Gandhinagar, Roll No: 20110178

**Saatvik Rao**
Indian Institute of technology Gandhinagar, Roll No: 20110175

## Abstract

Bitcoin is a decentralized digital currency that was introduced in 2009 by an anonymous individual or group known as Satoshi Nakamoto. It is built on a blockchain technology which is a decentralized, distributed ledger that records transactions in a transparent and secure manner. Bitcoin operates without a central bank or single administrator, making it a peer-to-peer digital currency. Transactions are verified by network nodes through cryptography and recorded on a public ledger called a blockchain. The supply of bitcoin is limited to 21 million, with approximately 18 million currently in circulation. Bitcoin has gained popularity due to its decentralized nature, the potential for anonymity in transactions, and its deflationary monetary policy. However, it has also been associated with illegal activities such as money laundering, drug trafficking, and cybercrime due to the perceived anonymity of the transactions. As a result, there has been an increasing need for monitoring and regulation of the cryptocurrency industry. This has led to the development of various tools and techniques to track and analyze blockchain transactions, including the use of data science and machine learning. The aim of this data science project is to extract and classify addresses from the Bitcoin blockchain into four categories: criminal, exchange, miner, or service. The data will be collected from the blockchain and analyzed using various datascience techniques.

**Github Organization:** Link

**2012 ACM Subject Classification** Data Science

**Keywords and phrases** blockchain, on-chain data, evaluation data, API, multiprocessing, Graphsense, user identification, transaction analysis, data processing.

**Digital Object Identifier** 10.4230/LIPIcs.IITGN.2023.1

## 1 Background

Bitcoin transactions are recorded on a decentralized electronic ledger called the blockchain. Each user can have multiple public addresses and corresponding private keys. These addresses are used to send and receive bitcoin transactions.

The blockchain is a series of blocks containing information about each transaction that has occurred on the network. These blocks are linked together in a chronological order and cannot be altered, ensuring the integrity of the transaction history.

Since the Bitcoin network is peer-to-peer, there is no central server that stores all user balances. Instead, to calculate a user's balance, the algorithm traverses through all the blocks in the blockchain and calculates the balance based on the transactions associated with each public address.

To make this calculation successful, Bitcoin uses a full transaction scheme. In this scheme, if you want to send 5 BTC to your friend and you have 6 BTC in your wallet, you would create a transaction for 6 BTC in which 1 BTC is sent back to yourself and 5 BTC is sent to your friend's public address.

This transaction is then broadcast to the Bitcoin network where it is verified by network nodes using cryptography. Once verified, the transaction is added to a new block on the blockchain, and the balance of each public address involved in the transaction is updated accordingly.

**Problem Statement**
Fetch the bitcoin blockchain data, anlalyze the bitcoin network and classify user address as suspicious user, service, exchange or miner. Also set up evaluation setup for the model.

# 2 Methodology

## 2.1 Fetching Data

For the project at hand, two datasets are required - onchain data and evaluation data. Obtaining onchain data can be a time-consuming and computationally heavy process, so there are two methods available to retrieve it.
The first method involves synchronizing with the bitcore network and downloading block data from other users. This method requires downloading a massive 400GB of data and verifying each transaction, making it a computationally heavy process. The second method involves using an available API service to retrieve the data. One such API service is provided by blockchain.com, which allows users to obtain full block data by providing the blockhash. However, the data retrieved is in JSON format, and needs to be parsed before it can be used. For the current project, the team retrieved onchain data using the blockchain.com API service. They retrieved data for the month of March 2020, during which approximately 4320 blocks were added to the blockchain. This amounted to approximately 6-7GB of data. However, processing this data sequentially would take a considerable amount of time. Therefore, the team utilized multiprocessing and created 30 processes to fetch data for all 30 days of the month. This approach allowed them to save the data in .csv format, making it easier to work with.
Evaluation data is also required for the project, and this was obtained from an open-source organization called GraphSense. GraphSense collects data about which addresses are used for which purpose, and their sources include different agencies, forensics, and labs. They have a Git repository where they share their data in the form of 30-50 .yaml files, each containing different cryptocurrencies and their addresses.
The team parsed all these files and saved the data in .csv format, allowing them to integrate it with the onchain data obtained from the blockchain.com API service. The evaluation data is used to classify addresses as either criminal, exchange, miner, or service, which is the main objective of the project.
In conclusion, obtaining onchain data for a data science project can be a time-consuming and computationally heavy process. However, utilizing available API services and multiprocessing techniques can make the process more manageable. Evaluation data can also be obtained from various sources, such as open-source organizations like GraphSense, which can aid in the classification of the obtained onchain data.

| *transactionid* | *publicaddress* | *valueinsatoshi* |
|---|---|---|
| 789012 | 2AbCdEfGhIjKlMnOpQrStUvWxYz | 10000 |

**Table 1** Input transaction .csv file format.

| *transactionid* | *publicaddress* | *valueinsatoshi* |
|---|---|---|
| 189082 | 2cfddEfGhIjKlMnOpQrStUvWxYz | 103200 |

**Table 2** Output transaction .csv file format

| *publicaddress* | *tag* |
|---|---|
| fdsddEfGhIjKlMnOpQrStUvWxYz | criminal |

**Table 3** Evaluation data format

## 2.2 Analyzing data

In order to analyze the data, the first step is to assign addresses to user IDs. This process is necessary because one user can have multiple addresses. To address this issue, a heuristic can be used. If a user uses multiple addresses in a single transaction, it is assumed that the user has access to the private keys of those addresses. Thus, if multiple transactions are seen from a single address, it is likely that the address belongs to a single user. By using this heuristic, the data can be traversed, and each address can be assigned a user ID. If two addresses have the same user ID, it means that both addresses belong to the same person.
Once the user IDs have been assigned, various data points can be extracted, such as the average number of transactions per day, the average number of degrees (degree refers to the number of distinct users that a user transacts with) per day, and the frequency of transactions. Additionally, the user can be assigned a tag (criminal, miner, exchange, service) using data fetched and processed from Graphsense.
To handle the large amount of data, multiprocessing or multiple threads are used. Accessing the hard disk frequently adds delay to the process, but using threads can help fill the waiting time with other processing. By using these techniques, the data can be efficiently processed and analyzed..

You can see data in Figure 1a,1b,2a,2b, and 3a. Here cumulative average for example if user 1 has some attribute value 2, user 2 has value 5, user 3 has value 4 then plot will be (user1,2),(user2,3.5),(user3,3.33) x axis shows the user id and y axis is fetched feature. In Figure1 and Figure2 scales are logarithmic scale.

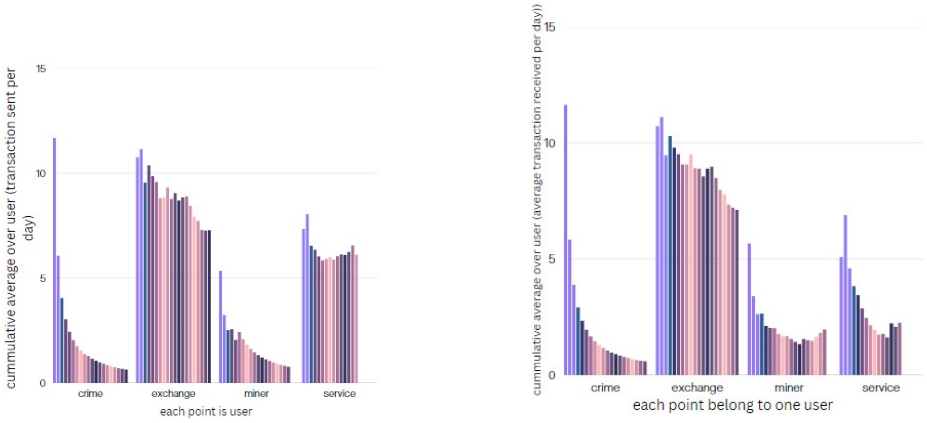
**Definition 1 Average transaction sent per day** is equal to total transaction sent in a march month divide by number active days.

**Definition 2 Average transaction received per day** is equal to total transaction received in a march month divide by number active days.

**Definition 3 Average out degree per day** is equal to average number of different users received amount from particular user.

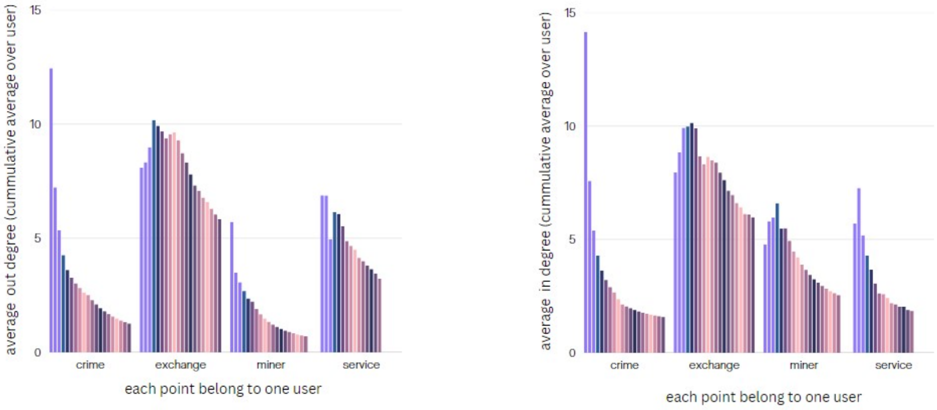
**Definition 4 Average in degree per day** is equal to average number of different users received amount from particular user.

**Definition 5 frequency** is equal to total number of active days divide by difference in days between last active day and first active day.



**(a)** **(b)**

**Figure 1**



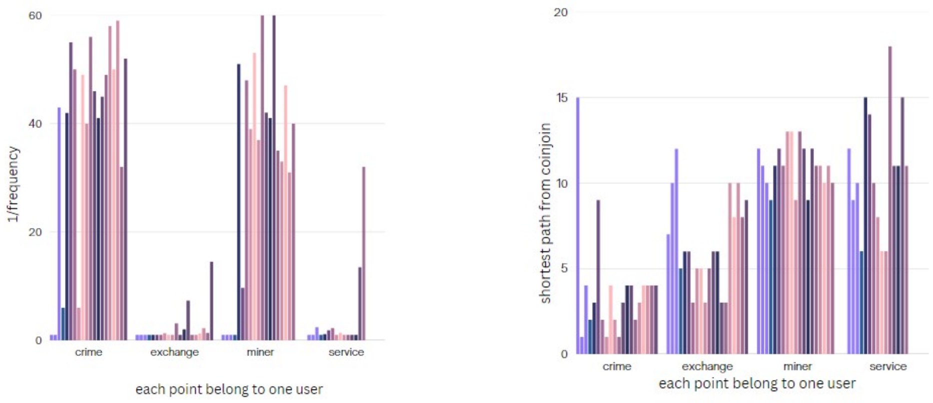
**(a)** **(b)**

**Figure 2**

Here we can see that miner and suspicious user are inactive and their transactions per day and degrees are less than other users. So, these feature will help us identify or differentiate between miner and exchange, crime and service and other combinations.

In bitcoin network there exists some services which will make you totally anonymous we can't track users after they uses these services. Mainly these services called coin join services. They take your current address and new address as input and transfer all your bitcoin from older to newer one for you. they can do this by multiple transaction or single transaction by gathering lots of users. But there is fee for writing transaction in data block these fees goes to miner and amount depends on transaction size. So, these services use optimization techniques to decrement their cost and now we can find some patterns using existing heuristic[2]. There are 2 popular services Wasabi and Samourai.

Wasabi CoinJoin Detection Heuristic (WCDH) If a transaction t has at least ten equal



**(a)** **(b)**

**Figure 3**

value outputs, with 0.1 ± 0.02 BTC being the most frequent one, and if it has at least three distinct output values with at least one being unique, and if it features at least as many inputs as occurrences of the most frequent output, then it is a Wasabi Wallet CoinJoin transaction.

Samourai CoinJoin Detection Heuristic (SCDH) If a transaction t has exactly five uniform outputs that equal p BTC and if it has precisely five inputs, with at least one and at most three equal p BTC, while the remaining two to four inputs are between p and p + 0.0011 BTC, then t is a Samourai Whirlpool CoinJoin Transaction.

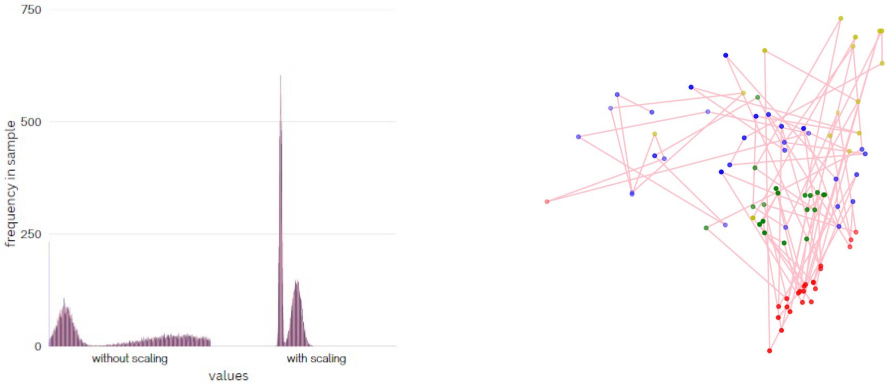
Now we can go through all transations and find out which transation is coinjoin transaction. Finding these transaction will help to improve our model because suspicious users use these services more frequently than others so we can guess that shortest path from coinjoin transaction for suspicious users will be lower. you can see plot for this data in figure 3b.

### 2.3 Scaling

We are applying kmeans. when we were applying kmeans algorithm we realized that that the in the data there were many outliers so, kmeans algorithm was giving centers to father points and accuracy was unsatisfactory. So, we applied interquartile range scaling. In Figure 4a you can see before scaling data and after scaling data. the data is not actual data its generated data to explain results.

## 3 Results and Conclusion

You can see data in Figure 4b after applying svd. We applied kmeans algorithm with 8 number of clusters. And we got good results we got 82 percent accuracy. What we did is we applied kmeans and we got the clusters after that we are saying this cluster consists of all criminals if the most users in that clusters are criminals. We idetified all suspicious user correctly except one.
In conclusion, our data science project involved analyzing blockchain data to gain insights about user behavior and identifying potential illicit activities such as money laundering. We



**(a)**

**(b)** crime(red), exchange(blue), miner(green), service(yellow)

**Figure 4**

used two datasets, on-chain data and evaluation data, and applied various heuristics to assign user IDs and tags to addresses.

By analyzing the data, we were able to extract valuable information such as the average transaction per day, the average number of degrees per day, and the frequency of transactions. We were also able to identify potential criminal activity through the use of specific tags.

To process the large amount of data, we utilized multiprocessing and multiple threads to improve efficiency and reduce waiting time.

Overall, this project demonstrated the importance of data analysis and its ability to provide valuable insights into complex systems such as blockchain. By identifying patterns and trends in the data, we can better understand user behavior and potentially prevent illicit activities.

## 4 Future Work

We had limitation that we cann't download all the data do analysis. But with decent processing machine this can be done with our code. And if we have all the data then we can extract more features like life of the address, etc. Also we will have more training data.

## References

[1] Y. Zhang, J. Wang and J. Luo, "Heuristic-Based Address Clustering in Bitcoin," in IEEE Access, vol. 8, pp. 210582-210591, 2020, doi: 10.1109/ACCESS.2020.3039570.

[2] Stockinger, Johann Haslhofer, Bernhard Moreno-Sanchez, Pedro Maffei, Matteo. (2021). Pinpointing and Measuring Wasabi and Samourai CoinJoins in the Bitcoin Ecosystem.

[3] S. X. Wu, Z. Wu, S. Chen, G. Li and S. Zhang, "Community Detection in Blockchain Social Networks," in Journal of Communications and Information Networks, vol. 6, no. 1, pp. 59-71, March 2021, doi: 10.23919/JCIN.2021.9387705.