# Offline Detection of Influential Bitcoin Users

Samuel Colbran (samuco@stanford.edu), Jake Smola (smola@stanford.edu)

## 1. Introduction

Bitcoin is a worldwide cryptocurrency launched in 2009 that enables distributed peer-to-peer digital payments without a central authority. The network uses a blockchain (commonly referred to as a public ledger) maintained by 'miners' to record transactions. The miners prevent tampering through the idea of proof-of-work: an exponentially increasing amount of computational processing power must be spent before a block of transactions can be added to the blockchain. The usage of this blockchain makes the Bitcoin network an ideal candidate for study, as all of the information is publically available and past entries cannot be removed. This means that a record of every single (successful) Bitcoin transaction can be downloaded and we can be confident that no components of the network are missing.

### 1.1 Motivation

We wish to explore the Bitcoin transaction network in order to find potential influential nodes and characterize the relationships between clusters, supernodes or entities. For example, where are the mining pools sending their money after they create it? Are they pulling it out of the Bitcoin network by making large or regular transactions with an exchange, or are they keeping it in the network? This type of work may also suggest additional ways to label previously unlabeled clusters. For example, if a user interacts with an exchange, that exchange likely obtains personal financial information from them (which is especially relevant when talking about illicit activities). Previous attempts to solve this problem have required live, on-net operations on the Bitcoin network, which is slow, requires large bandwidth, is easily detected by everyone and in general requires crawling over the network (not to mention the online crawler might be blacklisted for spamming nodes). It also misses nodes that perhaps are very influential and simply offline (maybe they only operate once a week) or are not connected to a point at which the crawler starts. In any case, we believe at least a subset of the prior results may be ascertainable in an offline analysis of existing Blockchain transactions alone, or may reduce the resources required by an online approach to find influential nodes.

### 1.2 Problem Definition

We aim to determine the most influential entities within the Bitcoin network using only the offline transactional data. The scope of our analysis is focused on using the preprocessed dataset provided by ELTE (see 3.1 Data Collection).

## 2. Related Work

The concept of influential bitcoin nodes has been addressed or proposed in several pieces of literature.

### 2.1 Discovering Bitcoin's Public Topology and Influential Nodes

Miller et al. [13] introduce a variety of techniques they conceived in uncovering the disproportionate distribution of power in the Bitcoin network. These techniques include AddressProbe, a method for discovering peer-to-peer links in the Bitcoin network, Candidate Selection, an algorithm for finding potential influential nodes, and Influence Validation, an algorithm that confirms the excess (mining) power of selected nodes. Using these techniques, Miller et al. are able to directly identify influential nodes that wield disproportionate mining power in the Bitcoin network. The authors model the Bitcoin network as a collection of nodes and edges, then apply graph theoretic measures and methods thereto, in order to compute node-degree and connectedness and also make an attempt at node coloring. This paper assess graph topology, seeks to gain additional insight from said topology and in particular applies this concept to the Bitcoin network.

### 2.2 Characterizing payments among men with no names

In [12], Meiklejohn, et al. discuss clustering Public Keys / Bitcoin wallets based on transaction history and attempt to determine what each cluster is doing or who they are by making multiple transactions with mining pools, exchanges, vendors, gambling sites, etc. They discuss basic statistics and show that certain idioms of use are likely to expose users and should thus be avoided by platforms using Bitcoin. In addition to a small set of labeled nodes achieved by interacting with a bunch of services (and thereby labeling a single key as e.g. "silk road"), they were able to label a large fraction of the active Bitcoin public keys.

The paper uses two heuristics; a standard "inputs to a single transaction are owned by the same source" heuristic that had been explored before, and a newer heuristic about the use of change addresses to split up a large received transaction into more workable smaller transactions to be used as inputs to future transactions and clustering together

any keys involved in such transactions. They also explore several practical consequences of the clustering, which we won't discuss, and conclude that, practically, Bitcoin fails to achieve the anonymity attributed to it by advocates.

### 2.3 Other Related Contributions

Several intersecting pieces of work done in the realm of Bitcoin, abuse, and influence are available in [6], [11], [8], [9], [15], [16], and [18].

## 3. Method

### 3.1 Data Collection

We have elected to download the Bitcoin dataset provided by the Hungarian Virtual Observatory (HVO) at the Eötvös Loránd University (ELTE) [7]. They provide the entire blockchain up to Oct 19, 2014 containing 326,027 blocks, and a separate computed version that contains "possible identification of addresses belonging to the same user and the reconstructed directed graph between the addresses." We could alternatively download the entire blockchain (using a program such as Bitcoin Core) from a peer within the live Bitcoin network, but the preprocessing provided by HVO allows us to focus more on analysis without worrying about the structured Bitcoin data formats. A bit of preprocessing was still necessary to convert the provided data into a format that the Stanford Snap library could read.

### 3.2 Background

We model a graph using the notation $G = (V, E)$ where $V$ represents the set of nodes and $E$ represents the set of edges. $|V|$ is the number of nodes in $G$ and $|E|$ is the number of edges. In much of our discussion, we are be concerned primarily with directed graphs, and thus each edge $E$ consists of a source node and a destination node and the direction of this relationship is meaningful.

For this project, we are considering the transaction network and the corresponding user network which we can deduce from transactions alone. Each transaction contains the participants' bitcoin wallet address; however, these addresses are cryptographically generated to preserve the anonymity of the user (entity) behind each address. Note that every transaction in bitcoin can have multiple input and output addresses as demonstrated in figure 1. In the transaction network, each node $i$ represents a bitcoin address and each edge $(i,j)$ represents one or more transactions with input address $i$ and output address $j$. Note that for simplicity, the dataset maps the actual bitcoin addresses into a continuous space of integers. When we refer to an *address* in this paper, we are describing the custom label and not the real address associated with some behavior.

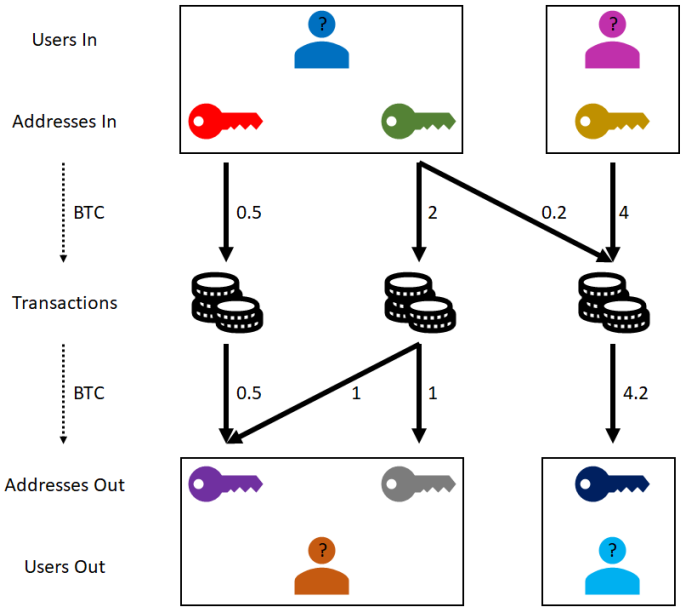



*Figure 1: Bitcoin transaction schema*

## 3.3 Models & Algorithms

In this section, we briefly survey a variety of classical algorithms that we have implemented in solving our problem. We provide these algorithms as a reference to support future work in this field.

### 3.3.1 Bow Tie Model

Broder et al. [6] present the model shown in figure 2. The 'IN' component includes nodes with a path following the directed edges into the strongly connected component (SCC). The 'SCC' also contains directed edges into the 'OUT' component, but not vice-versa. The last component is the 'TENDRIL', where nodes contain directed edges to either the 'IN' component or from the 'OUT' component (without a self path).

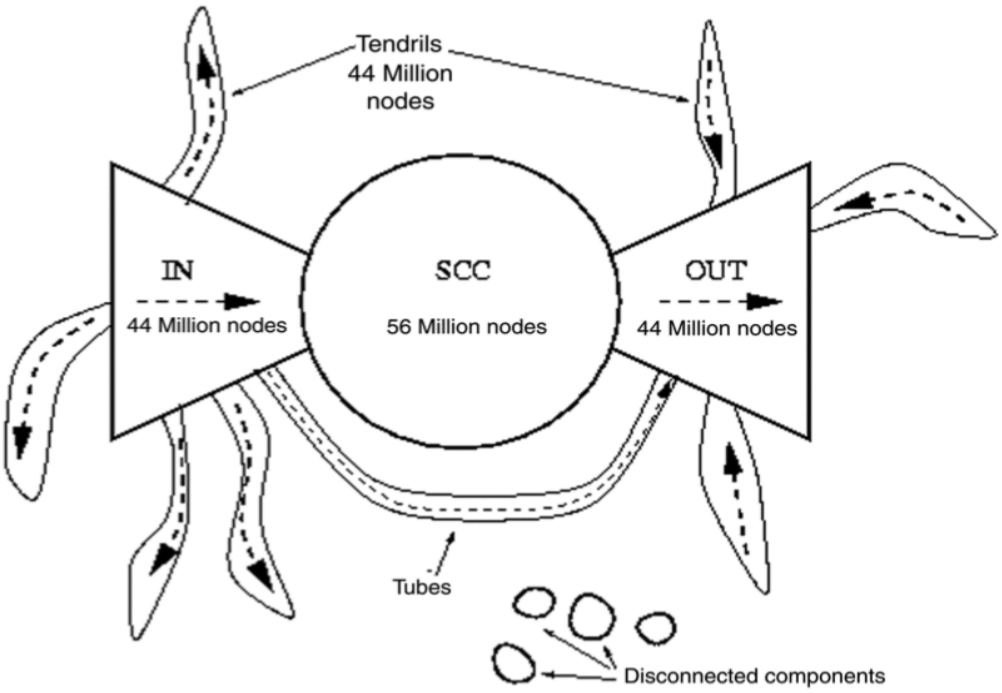



*Figure 2: Broder et al. Bow Tie Model [6]*

### 3.3.2 Breadth First Search Algorithm

We use breadth first search (BFS) to classify nodes according to the bow tie model. BFS starts from a source node and considers all subsequent neighbors in order of their distance from the start node. As an example, BFS first considers the source node (distance 0), then immediate neighbours (distance 1), then neighbours of neighbours (distance 2), etc. In our construction of the bow tie model, we leverage BFS to move from node to node and deduce the component of each node based on its local structure with other, classified nodes.

### 3.3.3 Average Clustering Coefficient

Recall that the local clustering coefficient is defined as

$$C_i = \frac{2|e_i|}{k_i \cdot (k_i - 1)} \text{ if } k_i \geq 2 \text{, otherwise } C_i = 0$$

where $k_i$ is the degree of node $i$ and $e_i$ is the number of edges between all neighbors of $i$.

Intuitively, this measure communicates how connected the neighbors of a particular node $i$ are. The *average* clustering coefficient is just the normalized sum of all local clustering coefficients in the graph:

$$C = \frac{1}{|V|} \sum_{i \in V} C_i$$

### 3.3.4 Eigenvector Centrality

Eigenvector centrality, first introduced in [3], provides a measure of influence of each node in a network. Eigenvector centrality takes $G$ and scalar parameter $\lambda$ as input and can be computed using power iteration as follows:

1. Set the influence at time 0, $r_j^{(0)}$, of each node $j$ to be $\frac{1}{|V|}$.
2. Set time $t = 1$
3. Until convergence ($\Sigma_j |r_j^{(t)} - r_j^{(t-1)}| > \epsilon$ for some $\epsilon$):
    a. For every node $j$:
        i. Set $r'_j{}^{(t)} = \frac{1}{\lambda}\Sigma_{i \to j} r_i^{(t-1)}$
    b. For every node $j$:
        i. Set $r_j^{(t)} = r'_j{}^{(t)}$ and normalize.

The influence of neighboring nodes plays a large role in determining each node's influence, and we can see that eigenvector centrality tends to associate influential neighbors with influential nodes.

### 3.3.5 Betweenness Centrality

Betweenness centrality was first introduced in [1]. The betweenness centrality of a given node $i$ is defined as the fraction of shortest paths between any two vertices other than $i$ that pass through $i$. We can express this as follows:

$b_i = \sum_{i,s,t \in V} \frac{\sigma_{st}(i)}{\sigma_{st}}$ where $s \neq i \neq t$, $\sigma_{st}(i)$ is the number of shortest paths between $s$ and $t$ that pass through $i$, and $\sigma_{st}$ is the total number of shortest paths between $s$ and $t$.

There are several ways to implement this algorithm, some more obvious than others. Our underlying implementation leverages a fast yet less intuitive alternative whose details we will spare. Further information on this improved algorithm can be found in [4].

### 3.3.6 Page-Rank Algorithm

Page-Rank [5] provides insights similar to Eigenvector Centrality; however, unlike Eigenvector Centrality, Page-Rank accounts for edge direction. This is particularly useful in our case where we are working with a directed transaction network.

The algorithm takes $G$ and scalar parameter $\beta$ as input and proceeds as follows:

1. Set the rank at time 0, $r_j^{(0)}$, of each node $j$ to be $\frac{1}{|V|}$
2. Set time $t = 1$
3. Until convergence ($\Sigma_j |r_j^{(t)} - r_j^{(t-1)}| > \epsilon$ for some $\epsilon$):
    a. For every node $j$:
        i. If in-degree of $j$ is 0: set $r'_j{}^{(t)} = 0$
        ii. Else: set $r'_j{}^{(t)} = \Sigma_{i \to j} \beta \frac{r_i^{(t-1)}}{k_i}$ where $k_i$ is the degree of node $i$ and $i$ is an in-neighbor of $j$.
    b. For every node $j$:
        i. Set $r_j^{(t)} = r'_j{}^{(t)} + \frac{1 - \Sigma_j r'_j{}^{(t)}}{|V|}$
    c. Increment $t$

# 4. Results

## 4.1 Summary Statistics

### 4.1.1 Unique Address Graph

The unique address interaction graph (UAI) is one where a directed edge is created from address A to address B if A and B have been involved in a transaction where A is an input and B is an output. Under the assumption of the bow-tie model (as discussed previously), UAI has the following properties.

*Table 1: Unique address graph statistics*

| | | |
|---|---|---|
| Nodes = 24575385 | SCC = 21655997 | IN = 1066595 |
| Edges = 89220163 | WCC = 24573384 | OUT = 1816833 |
| Tendrils = 33959 | Disconnected = 2001 | Avg. Clustering Coefficient = 0.105593219716 |

### 4.1.2 Unique User Graph

The unique user interaction graph (UUI) is one where a directed edge is created from user A to user B if user A has been involved in a transaction where an address owned by A is an input and an address owned by B is an output. The preprocessed map of address to user provided by HVO was used during the creation of this graph. UUI has the following properties.

*Table 2: Unique user graph statistics*

| | | |
|---|---|---|
| Nodes = 12094227 | SCC = 9630074 | IN = 739714 |
| Edges = 33247912 | WCC = 12093152 | OUT = 1695910 |
| Tendrils = 27454 | Disconnected = 1075 | Avg. Clustering Coefficient = 0.134870765402 |

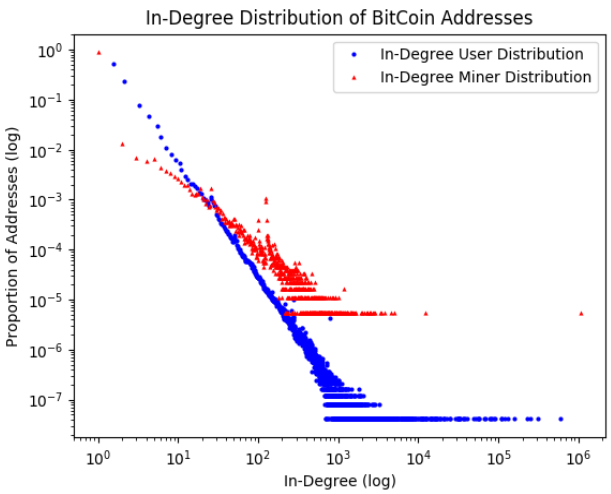
## 4.2 Miner Discovery

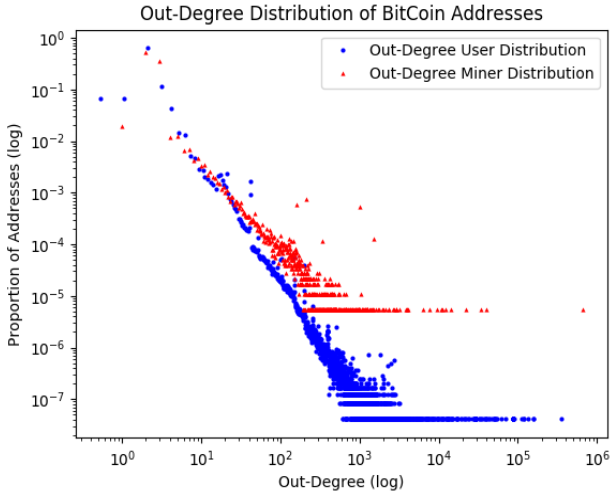
We were able to label a set of nodes as ‘miners’ by investigating transactions and the total sum of their inputs and outputs. Nakamoto, S. [14] explains that users wishing to have their transaction included on the blockchain require that a miner adds it to a block and then successfully computes the proof-of-work before anyone else. As there are many transactions, it is in the interest of the miners to pick transactions with additional incentive such as a reward payment or transaction fee, which is paid to the miner if they successfully add the transaction to a new block in the blockchain. Users are essentially bidding on the right to have their transaction included in the next block. We saw these cases while investigating the transactions:

1. If *input* = *output*, the transaction is balanced.
2. If *input* > *output*, bitcoin is disappearing. This case occurs more often today than case 1, as the missing ‘bitcoin’ from the input ends up in the output of a separate case 3 transaction that is paid to the miner, in addition to a set amount of bitcoins rewarded from mining a new block (which started at 50 bitcoins and halves every so often).
3. If *input* < *output*, bitcoin is being generated or taken from another transaction. We can exploit this structure to identify the “miner reward” transaction within each block and identify a set of addresses that are most likely miners.

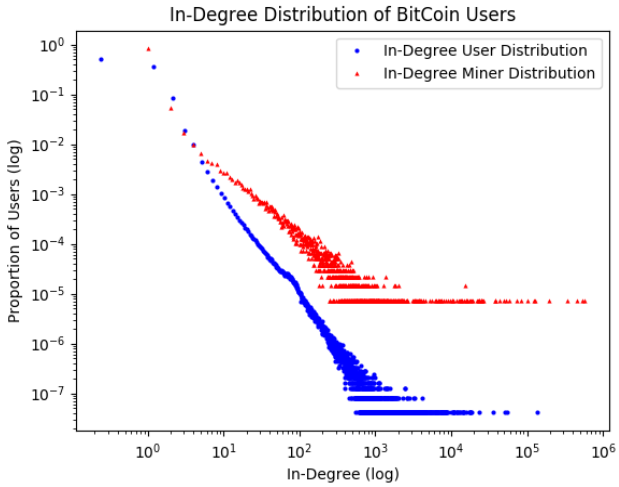
It should be noted that this method does not identify all of the miners, as more complicated graph structures designed for anonymity may be involved. For example, a miner pool may pay all of the dividends to a set of temporary addresses, which then distributes them to members of the mining pool at a later date. A variety of these tactics, such as Long Chains, Fork-Merge Patterns, Self Loops, Binary Tree-Like Distributions etc. are discussed further by Ron et.al in [18]. It is difficult to label miners in this case and we need to rely on other methods, as described in later sections. The other downside to this approach is that it does not capture miners that are failing to mine; though presumably miners with such small compute power would join a mining pool.

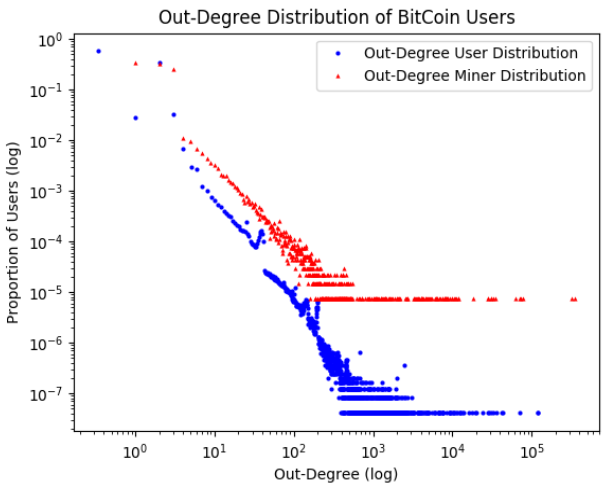
Using this immediate labelling, we can split the degree distribution for both users and addresses into ‘miners’ or ‘users’ which both appear to follow a separate power-law distribution as shown in figures 3, 4, 5, 6. The user graph in particular shows a very clear separation, and the results indicate that miners tend to have a higher proportion of transaction degrees than their non miner counterparts, which may be explained by the distribution of wealth. In the network, every single bitcoin must originate from a miner. It is therefore only natural for miners to be involved in more transactions than their non-miner counterparts, as they start with more money and therefore have a greater potential to spend. It is also possible that the proportionally higher miner degrees are caused by the payment of transaction fees.




*Figures 3, 4: In-Degree and Out-Degree of BitCoin Addresses respectively for Miners vs Non-Miners*




*Figures 5, 6: In-Degree and Out-Degree of BitCoin Users respectively for Miners vs Non-Miners*

Kondor, et al. [10] also found that the transaction network for addresses followed a power-law distribution with in-degree exponent 2.18 and out-degree exponent 2.06, but did not cluster addresses into users or split the networks into miners vs non-miners.

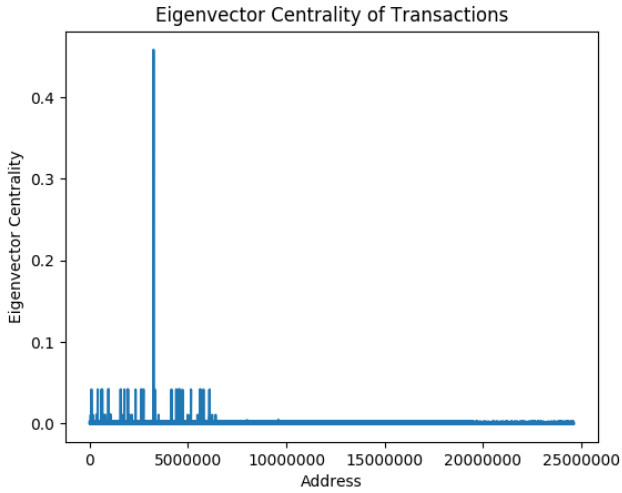
## 4.3 Influence Statistics

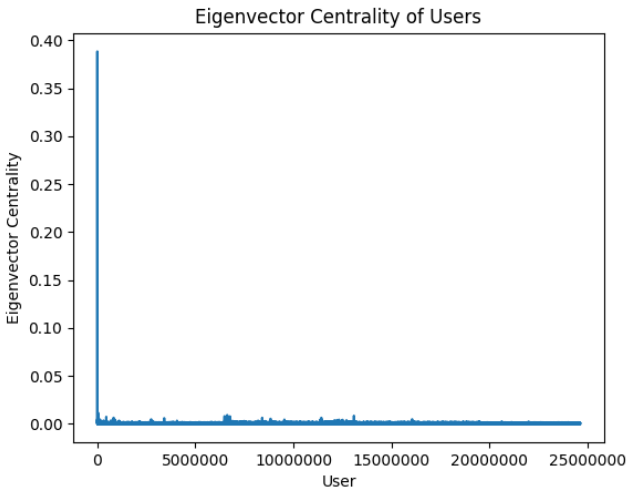
This section describes our initial findings with regard to the centrality of addresses and users in the bitcoin transaction network.

### 4.3.1 Eigenvector Centrality

The eigenvector centrality measures for the unique address and unique user graphs are plotted in figures 7 and 8. We can observe a significant outlier in the unique address eigenvector centrality data, where address 3,247,203 attained a maximum centrality of 0.458201 while the nearest runner-up attained a centrality of 0.113681. Closer inspection revealed this address was funded 1,082,605 transactions and benefitted from 669,606 transactions. Based on analysis in section 4.2, we have identified this address as a presumed miner, and it is the only miner in the top 50 addresses ranked by eigenvector centrality.

In the user graph, high eigenvector centrality again seemed to be concentrated on a single node. In this case, user 64 achieved the highest eigenvector centrality of 0.388319, with the runner up as user 98 with an eigenvector centrality of 0.099635. User 64 funded 578,783 transactions and benefitted from 355,156. based on miner analysis, was among *several* presumed miners with high eigenvector centrality. 35 of the top 50 identified users in terms of eigenvector centrality have been identified as miners (70%).



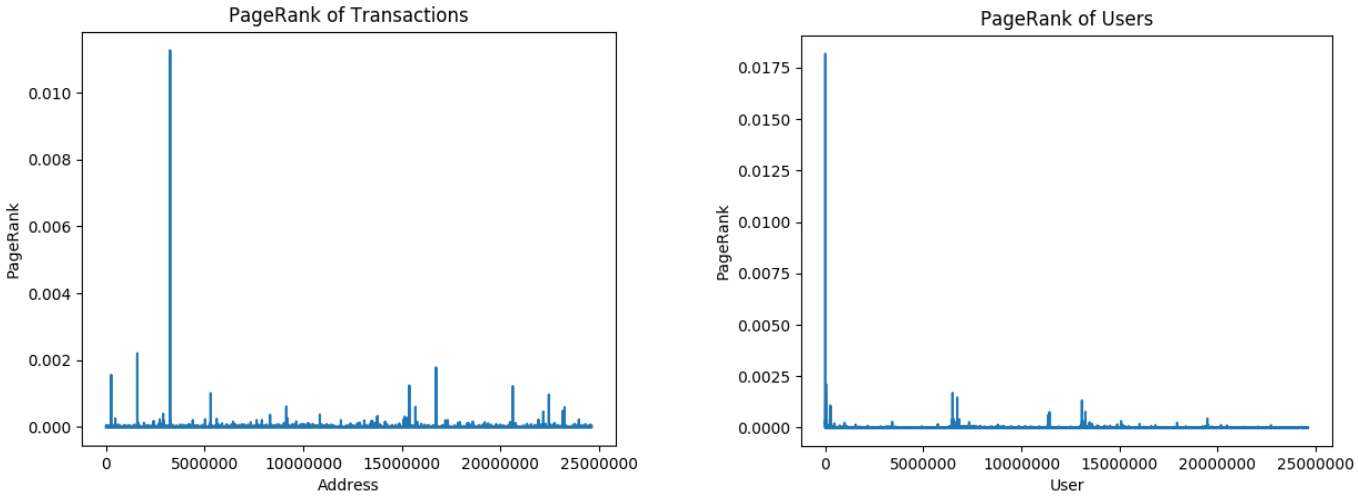




*Figures 7, 8: Eigenvector centrality in the unique address and unique user graphs*

### 4.3.2 PageRank

The PageRank measures for the unique address and unique user graphs are plotted in figures 9 and 10. Again, both plots suggest the existence of a few outliers. In the unique address eigenvector centrality data, address 3,247,203 attained the maximum PageRank of 0.011260 while the nearest runner-up attained a PageRank of 0.006476. This is the same address that achieved maximum eigenvector centrality (see section 5.2.1) and was identified as the sole miner among the top 50 nodes for eigenvector centrality.

The maximum PageRank in the user graph belonged to user 98 this time, with a PageRank of 0.018182, however, user 64 closely followed with a PageRank of 0.017371. User 98 was also identified as a miner and funded 528,512 transactions but benefitted from far fewer: 72,979. As in the case of eigenvector centrality, 35 of 50 top ranked users were identified as presumed miners. This begs the question of whether or not there is significant overlap between the top 50 users of both populations, or whether they complement one another; we answer this question in the next subsection.




*Figures 9, 10: PageRank in the unique address and unique user graphs*

### 4.3.3 Intersection of PageRank and Eigenvector Centrality

In the previous sections, we noted the equivalent proportions of top 50 users that were also presumed miners as ranked by PageRank and Eigenvector Centrality. Among the top 50 users under both models, 70% were identified as miners. However, when comparing the two user populations to one another, we found that only 48% of users from one population were present in the other. This means that slightly more than half of the 50 influential users each algorithm identified complemented the findings of the other algorithm. Furthermore, 9 of the top 10 nodes as classified by each algorithm were identical.

### 4.3.4 Address Ownership Identification

Directly identifying the wallet owner for a particular address requires investigation beyond the transaction network as there is no derivable mapping between the addresses used and the entities behind them. For example, the on-net methods utilized in [13] can be used to identify IP addresses associated with bitcoin relays that can provide information about the geographic disposition of wallet owners; however, still further analysis must be done to ascertain the identity behind the wallet. Our experiments cannot thus directly identify wallet owners, but we are interested in whether or not the nodes we deem influential do in fact coincide with influential entities. We thus compare our findings with a priori knowledge from [2], which maintains collected information about addresses and their owners.

The above results pertaining to bitcoin transactions are not nearly as interesting as the user results given the relatively low occurrence of address reuse (for the sake of anonymity). The average bitcoin user tends to take part in few transactions or uses multiple addresses across several transactions. Indeed, most of the influential addresses identified in the transaction graph belong to Satoshi Dice wallets. This is not surprising given the popularity of the service and its high frequency of address reuse. The user graph represents each user by a single node thus is more useful in finding influential entities.

For the user graph results of both eigenvector centrality and PageRank, we thus cross-referenced the top 10 addresses associated with each user using [2]. With this technique, we were able confirm several of the entities behind many of

these users is known to participate in bitcoin mining. Table 3 shows the wallet owners (if known) corresponding to our influential nodes. The table shows in bold the only wallets that are in the top 10 for *one* of the algorithms with their corresponding rank in according to the other algorithm in parentheses. PageRank recognized an influence similar to that of eigenvector centrality for “Bitcoin.de-old”; however, due to the wallet’s low degree (447 in, 34 out), eigenvector centrality failed to rank “00c8f6b5e82cab0e” among the top 50. We discuss this phenomenon further in the next subsection.

*Table 3: Top 10 influential bitcoin entities*

| Rank | Eigenvector Centrality | PageRank |
|---|---|---|
| 1 | MtGoxAndOthers | SilkRoadMarketplace |
| 2 | SilkRoadMarketplace | MtGoxAndOthers |
| 3 | BTC-e.com-old | BTC-e.com-old |
| 4 | Instawallet.org | SatoshiDice.com-original |
| 5 | Bitstamp.net-old | Instawallet.org |
| 6 | Bitcoin.de-old (12) | BitPay.com-old |
| 7 | 00005795a77580a4 | Bitstamp.net-old |
| 8 | SatoshiDice.com-original | 00005795a77580a4 |
| 9 | BitPay.com-old | 00c8f6b5e82cab0e (>50) |
| 10 | LocalBitcoins.com-old | LocalBitcoins.com-old |

### 4.3.5 Influence Discussion

Our influence findings thus extend the findings described in [13] by offering a complementary set of real-world entities that are influential in the bitcoin ecosystem. From the results above, we saw that influence derived from the transaction graph was not as telling as that of the user graph. This makes sense since users are encouraged to have multiple transaction addresses. Future analyses should thus focus more on user-oriented data. Additionally, we saw that PageRank was more evenly distributed among users than eigenvector centrality. The difference in these distributions afforded somewhat diverse sets of influential nodes that tended to complement one another. We believe both measures are thus useful in such experimentation.

However, if our experiment was limited to a single measure, PageRank would have been the preferred method. It is evident that unlike PageRank, eigenvector centrality is heavily influenced by node degree since it does not divide each centrality contribution by the degree of the source node. PageRank accounts for discrepancies in degree and thus identified influential nodes of lower degree; many of which were also presumed miners.

## 5. Challenges

### 5.1 Dataset Size

The size of the networks we considered introduced significant memory demands during experimentation. One of our primary challenges entailed running our initial experiments on commodity hardware (in some cases, an Intel i7 processor with 8GB RAM). This led to failed experiments which were preempted by memory failures or external user constraints. To mitigate this challenge, we applied optimizations by refactoring portions of pre-processing in c++ and ran some experiments on high-speed, dedicated compute nodes with 27GB RAM. We saw significant improvements in procedure runtimes on the dedicated compute nodes.

Unfortunately, our improvements were not sufficient to justify use of the newer bitcoin data, also available from [7]. We had originally intended to conduct our experiments with the newer dataset, which covers transactions up to Oct. 19, 2014. However, this dataset was twice the size of the dataset used in our experiments and would have consumed significantly more resources, limiting our ability to extract meaningful insights. We thus elected to use the older set, which covers transactions up to Dec. 28, 2013.

### 5.2 Betweenness Centrality Runtime

We initially sought to compare the betweenness centralities of users and transactions in order to gain insights regarding influence within each respective network. However, our first few experiments showed that the time complexity associated with achieving this goal was several orders of magnitude higher than that of eigenvector centrality or PageRank. We found this runtime unreasonable given our problem of identifying influential bitcoin users with limited resources.

We thus considered an approximate betweenness centrality algorithm as a cheaper alternative. The approximate betweenness centrality algorithm differed from the precise betweenness centrality algorithm in that it computed shortest path dependencies by only constructing breadth first search trees only from a random sample of nodes instead of all nodes. We assessed runtimes of the precise betweenness centrality, approximate betweenness centrality, eigenvector centrality, and PageRank on two networks: a network generated by randomized preferential attachment (5000 nodes of out-degree 4), a model that also demonstrates the power law characteristics we saw given our degree distributions, and the Epinions.com social network, a real network that models user-user interaction [17]. These networks were also directed but not as large as the bitcoin dataset we utilized; they thus proved useful for tractable runtime analyses.

We first established the baseline runtimes the three deterministic algorithms on the preferential attachment network. PageRank incurred the fastest runtime, while eigenvector centrality ran about 8 times slower than PageRank, and betweenness centrality ran about 4,900 times slower than PageRank. We used these run times to tune the node fraction parameter of the approximate betweenness centrality algorithm. We elected to reduce the runtime such that it became comparable to that of eigenvector centrality. Parameterizing on 1/1000 nodes achieved this goal.

We then computed the betweenness centrality of nodes in the Epinions network using the precise and approximate algorithms with our new parameter. From the output of each algorithm, we computed the top 50 nodes in terms of betweenness centrality. To our dismay, the approximate algorithm correctly identified only 15 of the 50 nodes that truly have highest betweenness centrality (30% accuracy). Further experiments sacrificed some time complexity for better accuracy, but we determined that it was not possible to achieve reasonable accuracy at a reasonable running time. We thus chose to discontinue pursuing betweenness centrality and focused instead on our other measures of influence and centrality.

## 6. Conclusion

Our aim for this project was to determine the most influential entities within the Bitcoin network using only the offline transactional data provided by the Hungarian Virtual Observatory (HVO). We started by finding several summary statistics, such as node / edge count, the size of each set according to the Bow-tie model, and average clustering coefficient. This gave us a good indication that the network was structured and not just a random graph. By exploiting the network structure, we were then able to figure out a set of users and addresses that belonged to miners. In addition, we plotted the degree proportionality for both the users and addresses network and discovered that they generally follow a power-law type distribution (interestingly enough, miners had higher degrees than their counterpart non-miners). Once we had a good sense for how the network was structured, we applied centrality measures, including eigenvector centrality, betweenness centrality, and the PageRank algorithm, in order to find the most influential nodes under each method. We encountered a few issues while attempting to calculate true betweenness centrality, primarily due to the size of each network and lack of compute resources. Unfortunately, an approximate algorithm yielded poor accuracy and we elected to discontinue pursuit of betweenness measures. With the remaining methods, we were nonetheless able to discover several addresses (the same address was often consistently ‘influential’ across different algorithms) that we were able to look up in [2]. This enabled us to confirm that each address was in fact “influential” - as they are owned by large exchanges or miners. The most influential turned out to be owned by the creator of Bitcoin, which is a very nice result. In conclusion, we achieved our goal to determine the most influential entities within the Bitcoin network using only offline transactional data and our results are consistent with the prior literature.

## 7. Future Work

In future iterations of this project, we would be interested in replicating this work with a more recent dataset--particularly one that captures bitcoin’s growing adoption and the recent bitcoin valuation surge among other perturbations. Such an analysis could potentially reveal other key players in the bitcoin ecosystem and newly influential nodes, many of which may also be associated with fraudulent activities.

Additionally, we would be interested in exploring alternative measures such as approximate pagerank and other optimizations that may improve runtime without significantly hindering accuracy. This would likely prove crucial to analysing the much denser, modern bitcoin transaction network.

## 8. References


[1] Bavelas, A. A mathematical model for group structure. Applied Anthropology, 7(3), 16–30, 1948.

[2] Bitcoin Whos Who. Available: http://bitcoinwhoswho.com/

[3] Bonacich P. Factoring and weighting approaches to clique identification. Journal of Mathematical Sociology, 2, 113–120, 1972.

[4] Brandes, U. A faster algorithm for betweenness centrality. *J. of Mathematical Sociology* 25(2), 163–177, 2001. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.11.2024&rep=rep1&type=pdf

[5] Brin, S. and Page, L. The anatomy of a large-scale hypertextual Web search engine. Computer Networks and ISDN Systems, 30(1-7), 107–117, 1998.

[6] Broder, A., Kumar, R., Maghoul, F. Raghavan, P., Rajagopalan, S., Stata, R., Tomkins, A., Wiener, J. *Graph structure in the Web.* Available: http://snap.stanford.edu/class/cs224w-readings/broder00bowtie.pdf

[7] ELTE Bitcoin Project Website and Resources. Hungarian Virtual Observatory Available: http://www.vo.elte.hu/bitcoin/downloads.htm

[8] Eyal I., Sirer E. G. Majority is not enough: Bitcoin mining is vulnerable. Financial Cryptography and Data Security, pages 436–454. Springer, 2014. Available: https://www.cs.cornell.edu/~ie53/publications/btcProcFC.pdf

[9] Gervais A., Ritzdorf H., Karame G. O., Capkun S. Tampering with the Delivery of Blocks and Transactions in Bitcoin. Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, October 12-16, 2015, Denver, Colorado, USA. Available: https://eprint.iacr.org/2015/578.pdf

[10] Kondor, D. Posfai, M., Csabai, I., Vattay, G. Do the Rich Get Richer? An Empirical Analysis of the Bitcoin Transaction Network. PLOS one, 2014. Available: http://journals.plos.org/plosone/article/file?id=10.1371/journal.pone.0086197&type=printable

[11] Lischke, M., Fabian, B. Analyzing the Bitcoin Network: The First Four Years. *Future Internet*. 2016. Available: http://www.mdpi.com/1999-5903/8/1/7

[12] Meiklejohn, S., Pomarole, M., Jordan, G., Levchenko, K., McCoy, D., Voelker, G. M., Savage, S. A fistful of bitcoins: characterizing payments among men with no names. Proceedings of the 2013 Internet Measurement Conference, IMC 2013, Barcelona, Spain, October 23-25, 2013 (2013), pp.127–140. Available: https://cseweb.ucsd.edu/~smeiklejohn/files/imc13.pdf

[13] Miller, A., Litton, J., Pachulski, A., Gupta, N., Levin, D., Spring, N., Bhattacharjee, B. Discovering Bitcoin's Public Topology and Influential Nodes. Available: https://cs.umd.edu/projects/coinscope/coinscope.pdf

[14] Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. Available: https://bitcoin.org/bitcoin.pdf

[15] Nayak, K., Kumar, S., Miller, A., Shi, E. Stubborn mining: Generalizing selfish mining and combining with an eclipse attack. IACR Cryptology ePrint Archive 2015 (2015), 796. Available: https://eprint.iacr.org/2015/796.pdf

[16] Ober M., Katzenbeisser, S.,Hamacher, K. Structure and anonymity of the Bitcoin transaction graph. Future Internet, 5(2):237–250, 2013. Available: http://www.mdpi.com/1999-5903/5/2/237/htm

[17] Richardson, M., Agrawal, R., Domingos, P. Epinions Dataset. Trust Management for the Semantic Web. ISWC, 2003. Available: https://snap.stanford.edu/data/soc-Epinions1.html

[18] Ron, D., Shamir, A. Quantitative analysis of the full Bitcoin transaction graph. Financial Cryptography and Data Security. Springer, 2013, pp. 6–24. Available: https://eprint.iacr.org/2012/584.pdf