# Bitcoin Network Analysis - Literature Review

The purpose of this document is to summarize the findings of our ongoing literature review process that has, up to date (03/10/2019), covered 28 articles published between 2011 and 2019, which analyze data from the the Bitcoin Network. The current list of articles and tools under review can be found here. Summary (draft) notes for each article reviewed can be found in the appendix of this document. Given our interest in graph analysis methods, this review is skewed towards work that applies these methods. This summary evaluates the literature from 3 perspectives: Objectives, Methodologies & Results and Tools.

## 1. Objectives

Literature related to Bitcoin Data analysis is centered around three main objectives: Presenting descriptive analytics of the networks and study its evolution; evaluate privacy guarantees, attempt to de-anonymize entities (users) and characterize their behaviour; create software or theoretical frameworks to better process data or model the network. Most of the articles reviewed correspond to the the second category (De-anonymization and Characterization). On this section we present the specific objectives pursued for each of these categories.

### *1.1. Analytics & Evolution*

Initial approaches like [2][6] focus on statistical analysis that summarize the transaction history of the network in its initial 4 years, measuring general network (transactions, addresses, balances and distributions) and graph (degree, centrality, distance, diameter) statistics, as well as their distributions. Some more recent work [15] has extended this analysis using a larger dataset and leveraging on better frameworks and computation capabilities to calculate other graph properties and their evolution [8]. Current research has also focused on analysing the evolution of the network's transaction and graph statics, comparing transaction data with external data (i.e. geographic, pricing, entity labels, forum sentiment analysis) [13][33] and showing that transaction behavior is consistent with traditional monetary theory [26]. Finally there is a line of work [16] that has focused on summarizing the main characteristics of existing blockchain technology as well as some research findings.

### *1.2. De-anonymization and Characterization*

Most of the existing literature is concentrated around the goal of exploring Bitcoin's privacy guarantees. Some work merely describes the state to which the network is truly anonymous [4] by applying basic clustering and deanonymization while other work takes a risk assessment perspective by presenting different de-anonymization techniques

[1][3], evaluating the effectiveness of exploiting vulnerabilities [10][20], or characterizing the lack of adoption of better anonymity preserving practices [23].

Another line of research that is concerned in applying specific 'ad-hoc' techniques to do forensic analysis around illicit network activity [5][7][11][21][31].

The most recent literature focuses on applying Machine Learning techniques to cluster addresses into users [22][9] or assign them to specific classes (exchanges, gambling, mining, etc.) [9][29][30].

### *1.3. Framework development*

A final line of research surveyed was focused on either developing software frameworks to parse/manipulate blockchain data [7][17][25] and visualize it [27] or define abstract formulations of blockchain structures [19] that might help deduce theoretical properties.

## 2. Methodologies & Results

The methodologies explored in existing research cover three main perspectives: 1. How are entities (users) clustered?; 2. How are their identities labeled? 3. How are their transactional relationships modeled? The main approaches presented in the literature for each of these perspectives are summarized below:

### *2.1. Clustering*

Users transact in the network pseudonymously by using bitcoin addresses of which they can create as many as they want. Ideally these addresses should only be used once in order to guarantee complete user anonymity. In addition, the fact that users have to reveal the addresses they control when they create transaction inputs or receive change, opens the possibility to cluster these addresses into users and explore their transaction patterns. A summary of clustering methods is shown below.

The most popular approach is to use what is known as the *common input heuristic,* which clusters addresses that are used in an input into single users. This process can be repeated transitively to cluster large groups of addresses shared by a common user [1][2][4][11][13]. The work done in [17] even shows a specific algorithmic implementation to apply this heuristic. Alternative approaches add to clusters (entities) defined by this first method a second heuristic that aggregates output addresses that are presumed to be the user's 'change' address, by identifying the number of decimals outputs [3] or the first-used address in outputs with only 2 addresses [5]. More current research uses (sometimes in addition to the previous heuristics) clustering techniques like the the community detection Louvain algorithm [22] or non-parametric tests to evaluate the behaviour of time series transaction behaviour [9]. Nonetheless most of the evidence suggest that the simpler common input heuristic is the most robust given that it requires less data and offers high precision and recall [28][22].

Some less common approaches are those that exploit specific vulnerabilities of certain tools that are used in the system. These include exploiting certain wallets that combine hashes and addresses in bloom filters [21] or that include change addresses in the first position of the outputs [7]. Although highly accurate, these methods are only effective for a small subset of addresses.

### *2.2. Labeling*

The procedures mentioned above are useful to cluster anonymous addresses into pseudonymous entities, nonetheless in order to categorize them or tie them to real world identities, it is necessary to find ways to label them.

Initial approaches used resourceful techniques to tie bitcoin addresses to real-world identities, such as scrapping public sources of data (forums, faucets) [1][7][11] or directly transacting with other entities in the network [5]. With time, services like Blockchain.info and Walletexplorer began aggregating these labels which has allowed other researchers [21][27][29] to scrape these services' websites or use their public-facing APIs. Nonetheless it is estimated that in 2018 less than 7% of all the bitcoin addresses could be labeled this way.

More recent work is using existing labels and address/user features to train machine learning models that categorize unlabeled addresses into categories such as *exchanges, mining pools, darkweb, gambling* or *other services*. Some models, capture node features such as its transaction history, network centrality or neighbor characteristics. These models have achieved F1 scores of 0.99 and 0.76 by training Random Forests [21] and Gradient Boosting Classifiers [30] respectively. Other models incorporate features of the different 1-2-3-motifs (subgraphs) that each address is involved in and reach F1 scores of up to 0.91 [29]. This last analysis identifies that features related to the 2-3-motifs are the most important in driving the effectiveness of the classifier.

### *2.3. Graph Modeling*

By construction, the Bitcoin blockchain defines a very specific relationship between addresses, transactions and blocks. Nonetheless the need to extract different types of data has lead researchers to select various graph structures to model this relationship. These different structures are presented below.

The most common graph structure used in the literature is a weighted directed multigraph, were the nodes are the identified users (entities) and an edge is defined between two nodes, *A* and *B* if there is a transaction that has as inputs an address associated with node *A* and as an output and address associated with node *B*. It is a 'multi'-graph because there can be more than one edge between two nodes and the edges are weighted depending on the value of the transaction. This structure is used by some researchers [15] to show that the transaction graph has a long diameter (2050 vs

Facebook's 34), but an average distance (4.5) and clustering coefficient (18) similar to other complex social networks. Other work shows how the network exhibits 'small world' phenomena [6][13] and how the unequal Gini coefficients (above 0.95) and power law distributions for in/out-degree centralities can be explained by preferential attachment [8].

Another approach, first proposed by [1], is to define a directed hypergraph between clusters of addresses without explicitly abstracting the entities defined by these clusters. This approach has allowed some authors to perform graph clustering algorithms on top of the address graph [22] or to explore complex subgraph topologies for forensic analysis [2]. Other approaches define bipartite directed graphs with entities or addresses as one type of node and transactions as another [5], allowing to calculate specific features for classification [29] or detect representative transactional motifs [21].

Less common approaches model the blockchain as a directed acyclic graph: [19] creates an abstract model for any type of blockchain model where the nodes are different states and the edges connect two neighboring states that are mediated by a transaction; [23] defines a directed acyclic graph connecting transactions to illustrate how information travel through out the bitcoin blockchain.

As an outlier we mention [27] that, for visualization analysis, defines a simple temporal graph that relates exchanges and the trade volume between them.

## 3. Tools & Data

This section enumerates the data sources and tools used in the literature reviewed.

### *3.1. Data sources*

- Blockchain.info: Online database of blockchain transactions and wallet (user) labels.
- Walletexplorer.com: Online database of blockchain transactions and wallet (user) labels.
- Amazon EC2 AMI running full node and BlockSci: https://github.com/citp/BlockSci
- BigQuery Public crypto Datasets: https://cloud.google.com/blog/products/data-analytics/introducing-six-new-cryptocurrencies-in-bigquery-public-datasets-and-how-to-analyze-them

### *3.2. Data extraction and structuring tools*

- BlockApi: Blockchain to database in Scala - https://github.com/blockchain-unica/blockapi
- BlockSci: Blockchain in-memory analysis (sans-database) - https://github.com/citp/BlockSci

- Bitcoin Transaction Network extraction: Python 2.7 Bitcoin parser - https://github.com/ivanbrugere/Bitcoin-Transaction-Network-Extraction
- BitIodine: Rust - Bitcoin parser with clustering - https://github.com/mikispag/bitiodine

### *3.3. Graph analysis and visualization tools*

- Webgraph: Java Framework (with Python interface) to handle large graphs - http://webgraph.di.unimi.it/.
- Snap: Network analysis framework C++ and Python - http://snap.stanford.edu/
- Neo4j: Graph Database - https://neo4j.com/
- NetworkX: Python network analysis tool - https://networkx.github.io/
- Pygraphviz: Python Graph visualization tool - https://pygraphviz.github.io/
- Gephi: Desktop graph visualization open-software - https://gephi.org/