Financial Innovation

RESEARCH

Open Access

Unsupervised clustering of bitcoin transactions

George Vlahavas[1*], Kostas Karasavvas[1] and Athena Vakali[1]

*Correspondence:
gvlahavas@csd.auth.gr

[1] Department of Informatics, Aristotle University of Thessaloniki, Thessaloníki, Greece

Abstract

Since its inception in 2009, Bitcoin has become and is currently the most successful and widely used cryptocurrency. It introduced blockchain technology, which allows transactions that transfer funds between users to take place online, in an immutable manner. No real-world identities are needed or stored in the blockchain. At the same time, all transactions are publicly available and auditable, making Bitcoin a pseudo-anonymous ledger of transactions. The volume of transactions that are broadcast on a daily basis is considerably large. We propose a set of features that can be extracted from transaction data. Using this, we apply a data processing pipeline to ultimately cluster transactions via a k-means clustering algorithm, according to the transaction properties. Finally, according to these properties, we are able to characterize these clusters and the transactions they include. Our work mainly differentiates from previous studies in that it applies an unsupervised learning method to cluster transactions instead of addresses. Using the novel features we introduce, our work classifies transactions in multiple clusters, while previous studies only attempt binary classification. Results indicate that most transactions fall into a cluster that can be described as common user transactions. Other clusters include transactions made by online exchanges and lending services, those relating to mining activities as well as smaller clusters, one of which contains possibly illicit or fraudulent transactions. We evaluated our results against an online database of addresses that belong to known actors, such as online exchanges, and found that our results generally agree with them, which enhances the validity of our methods.

Keywords: Bitcoin, Blockchain, Transactions, Clustering

Introduction

Bitcoin (Nakamoto 2008) is the first ever application of public blockchain technology. From its inception and first release in 2009, it has grown considerably and has inspired an explosive creation of other cryptocurrencies (Ballis and Drakos 2021). Many of these show small adoption, while others such as Ethereum (Buterin 2021) provide significant additional functionality and have also been adopted by an increasing number of people and organizations around the world. To date, Bitcoin remains one of the most popular and secure cryptocurrencies. For someone to launch a 51% attack against the Bitcoin network, so that they control all mining operations for one hour, they would currently need to spend more than $2B (Cost 2021), an amount is higher than any other

cryptocurrency. Hence, it would be practically infeasible to launch such an attack against the Proof-of-Work algorithm that is used to secure the Bitcoin network.

Transactions in Bitcoin do not explicitly identify senders or recipients of funds with their real-world identities. Instead, every user is identified by one or multiple pairs of cryptographic keys that make it possible to sign transactions and transfer funds from one public key to another, or equivalently, between bitcoin addresses. As details of all transactions, including their complete history, addresses, and public keys, are publicly available to anyone that connects a node to the Bitcoin network, Bitcoin is considered to provide a *pseudo-anonymous* service (Martins and Yang 2011).

Since all Bitcoin blockchain transaction data are publicly available, it is an open question to try to identify and classify users according to their usage patterns. Several attempts toward this goal have been made in published works, usually with the aim of identifying fraudulent or illicit behavior. Some use simple heuristics to classify transactions as non-standard (Bistarelli et al. 2019) and to deanonymize transactions related to CoinJoin mixers (Maksutov et al. 2019). Deanonymization of users by applying unsupervised learning techniques has also been examined (Hirshman 2013; Kang et al. 2020). Unsupervised clustering algorithms have also been explored to detect fraudulent or otherwise abnormal behavior (Lin et al. 2019; Nerurkar et al. 2020). In these cases, clustering is performed on addresses to detect entities and extract features; however, the final classifiers are based on supervised machine learning techniques. Indeed, one of the most common criticisms of Bitcoin is that it is often used for illegal activities such as money laundering, drug dealing, and funding terrorism (Xu 2016; Sicignano 2021). It is common for these works to base their findings on address clustering that is performed by applying certain heuristics on the transaction data, such as the reuse of addresses for multiple transactions.

The "traditional" way of managing bitcoin addresses involves the creation of a private key, which is essentially a very large random number. A public key is derived from this private key, which as the name suggests needs to remain hidden from others. The public key can be shared with anyone and with no restrictions. Each public key corresponds to a bitcoin address that can be used to receive funds. These funds can only be unlocked by the use of the respective private key. The generation of multiple addresses would require ownership and management of an equal number of private keys, which would be cumbersome. Hence, most users reuse the same addresses for their transactions as it is simply more convenient. A relatively recent advancement in this respect are Hierarchical Deterministic (HD) wallets (2021). Using this scheme, it is possible to derive an unlimited number of public keys, and therefore addresses, from a single private key, on the fly. The HD wallet standard is used by most wallet software released during the last couple of years. As a result, address reuse has drastically dropped in the Bitcoin network and up to 1,000,000 new unique addresses are currently recorded in the Bitcoin blockchain every day (Blockchain 2021). Owing to this advancement, the heuristics that used in several previous studies to cluster bitcoin addresses (e.g., Harrigan and Fretter (2016); Zhang et al. (2020)), may be rendered obsolete and makes identifying users by their addresses a complicated matter (Alqassem et al. 2020).

The sheer volume of blockchain transaction data makes machine learning a valuable tool for investigating patterns in such data, something that would not be as efficient,

or even possible, with more traditional techniques (Liu et al. 2020). Supervised learning techniques are usually more accurate than unsupervised ones; however, the lack of significant training data and labeled information make the use of supervised learning algorithms problematic (Ruppert 2004). This is very common in several financial applications, including fraud detection, credit evaluation and, reject inference (Li et al. 2022). Unsupervised learning methods do not require any training data to operate and it is possible to acquire useful results. Although the same lack of labeled information makes it difficult to accurately assess the performance of unsupervised learning methods, their results can be meaningful provided that features are carefully selected (Hinton et al. 1999).

The selection of features is a crucial step in the application of machine learning algorithms (Langley 1994). Choosing irrelevant features often leads to inconsistent clustering. At the same time, applying background knowledge is important for understanding the significance of resulting clusters (Danovitch and Keil 2004). In the absence of labeled data, only unsupervised clustering techniques can be used (Hinton et al. 1999). Previous studies that employed such methods mostly focused on classifying user activity as fraudulent or not.

The main purpose of address clustering in Bitcoin is primarily deanonymization, that is, discovering that different addresses are controlled by the same entities and therefore, by extension surmising the amount of bitcoins these entities hold. Combining this information with externally provided data, for example, knowledge that an address belongs to a specific entity like an online exchange, or an individual, may provide valuable information to actors such as attackers or law enforcement agencies. In contrast, the main purpose of clustering transactions in the Bitcoin network as we performed it is to group transactions with similar properties together, even if they belong to different entities, and characterizing these transactions according to their properties. Transaction clustering is therefore not an alternative to address clustering, but may be used together with the latter. It can potentially characterize an entity if its identity is otherwise unknown, or discover anomalous behavior between an entity's transactions if these are characterized in different ways. It has been demonstrated that working with financial transactional data can yield significant and interesting results (Kou et al. 2021).

Taking these into account, the contributions of this study include:

- The adoption of novel features that aid in the classification of Bitcoin transactions, requiring no external information other than the transaction data.
- Implementing a data processing pipeline that uses an unsupervised machine learning procedure to identify clusters of Bitcoin transactions, not limited to binary classification of transactions as illicit or not.
- The application of domain specific knowledge to characterize the resulting transaction clusters and therefore provide an overview of the kinds of transactions that occur on the Bitcoin network.

This rest of this paper is organized as follows. In the "Background" section, we introduce the basics of Bitcoin blockchain technology, dimensionality reduction and data clustering, defining concepts we deem necessary for understanding some of the technical

details presented herein. In the "Related work" section, we review existing literature on the subject. The "Methodology" section describes the data sources and methodology that we used to conduct our experiments. The results of our experimental study are presented in the "Experimentation" section. Finally, we set out the main conclusions in the "Conclusions" section.

Background

In this section, we present high-level background information with respect to how the Bitcoin blockchain works. This is necessary for understanding some of the technical details and decisions in the rest of the paper. A brief outline of the dimensionality reduction and clustering algorithms that were used is also provided for the same purpose.

Bitcoin

The Bitcoin blockchain is an immutable data structure that is shared over a peer-to-peer network between its users. As the name suggests, this data structure consists of a chain of blocks that are tightly linked to each other. Each new block that is generated is placed at the head of the chain. Every block always includes a unique identifier of its previous block. This unique identifier is a hash of the previous block's contents [or more accurately, a hash of the previous block's header contents using the SHA256 hashing algorithm (National Institute of Standards and Technology 2000)]. Since there is only one such identifier in each block, the blocks can be laid out in a straight line, from the first ever block that was generated, which is known as the "genesis" block, to the latest block, just by using the hash of the previous block to point to it each time. The immutable characteristic stems from the presence of these hash values. Since the hash value of any data changes unpredictably with any slight change to the data, it is impossible to alter any information stored in any of the previous blocks without altering the hashes stored in the next block, which in turn would alter the hashes of each next block and so on until the latest block. Therefore, any change to the information stored in the blockchain would be easily identified by anyone with a copy of the blockchain itself and rejected as invalid. The data stored on each block includes transaction information that moves bitcoins between accounts. Therefore, it is practically impossible for anyone to tamper with the transaction information stored in the blockchain.

New blocks are generated every 10 min, on average (Bonneau et al. 2015). The process of generating a new block is called "mining," as it is a very heavy and intensive process with respect to the computational resources it requires. Users that participate in this process are called "miners." The mining process involves the calculation of a valid hash value for the next block to be generated. For this hash value to be valid, it has to be smaller than a limit value; the hash value is just a (very long) number. Since the resulting hash value for any data is unpredictable, it is essentially a random number. Miners alter a predefined small part of the block, called the "nonce," which is reserved for this purpose, until they calculate a hash for the block that falls under the limit value. Since this is a random process, the more computational power a miner throws at the problem, the more likely they identify a suitable hash value for the block. This is what is referred to as the Proof-of-Work (PoW) mining process in Bitcoin. Once a suitable hash value has been calculated, it is broadcast to the network and all miners start working on mining

the next block. It is possible for two or more miners to find a suitable hash value at the same time and create a temporary fork in the blockchain. In such a fork, different parts of the network would think that a different block is at the head of the blockchain. However, situations like these are quickly resolved as it highly unlikely that this simultaneous generation of blocks will be repeated and bitcoin nodes always choose to follow the fork with the most blocks in it (or more accurately, the fork with the most work). In case the overall computational capability of the Bitcoin network is increased, blocks would be generated in times shorter that 10 min and the opposite is also true. The difficulty of producing a new block is adjusted every 2016 blocks by setting a different value for the limit that the hash values are compared against, therefore bringing the average block mining interval back to 10 min.

The miner that succeeds in mining the next block is rewarded an amount of bitcoins. These bitcoins are materialized at the same time as the mining of the new block occurs. This reward was equal to 50 bitcoins at the time that Bitcoin was launched. The computer code that Bitcoin nodes run specifies that this amount will be halved every 210,000 blocks, or roughly every four years. At the time of writing, the latest "halving" event took place on May 11, 2020, setting the reward from 12.5 to 6.25 bitcoins. In addition to the reward, the miner that succeeds in generating the next block also receives the sum of transactions fees that users have specified with each transaction that they submit to the network. The transaction that facilitates this reward attribution to the miner that managed to mine the new block is called a "Coinbase" transaction. Naturally, every block includes one and only one Coinbase transaction. In the rare event that two or more miners succeed in mining a next block at the same time, as previously described, all miners are awarded the mining reward, each one on their respective fork. As soon as the network stabilizes on which fork to follow, usually only after one or two more blocks are mined in one of the forks, the entire network switches to that fork and discards blocks in the competing forks, which are now orphaned. Therefore, any mining reward that was awarded in any of the failed competing forks is lost. To prevent miners from spending bitcoins that may effectively disappear if another fork prevails, the Bitcoin protocol dictates that miners are not allowed to spend their rewards for the next 100 blocks. The longest orphaned blockchain fork that has ever been created had a length of 31 blocks, from block 225,430 to block 225461. This occurred in March 2013 when there were fewer miners and their total hashing power was significantly smaller than it is today. It is highly unlikely that such a long fork will be created again. The next longest fork is only four blocks deep and occurred at block 174161.

Since mining is such a laborious process, and given there are a large number of miners trying to mine the next block, it is highly unlikely for each individual miner to succeed and therefore earn the reward for themselves. For this purpose, miners form groups, called "mining pools," that share resources and work collectively to mine the next block. If any member of the mining pool succeeds, then all members share a portion of the reward according to the rules set by each mining pool.

The bitcoin currency consists of smaller units called "satoshis," in collective homage to the creator of Bitcoin, Nakamoto (2008). Each satoshi is worth 0.00000001 bitcoins and is the smallest denomination of bitcoin. All transactions in the blockchain are actually stored and communicated as satoshis and only converted to bitcoins before display.

Bitcoins are transferred between users by submitting transactions to the network. With the exception of Coinbase transactions, every transaction includes some inputs and some outputs. The amount specified in all inputs always matches the amount specified in all outputs if the transaction fees are also included in the latter. Every time an output is specified with a new transaction, it is marked as "unspent," termed "UTXO" (Unspent Transaction Output). Every UTXO that is created remains unspent until it becomes an input for another transaction, in which case it is marked as "spent" and consumed in total. Multiple UTXOs can be combined to create outputs with amounts larger than any individual input. At the same time, multiple outputs may be specified if someone wishes to not send the entire amount of a UTXO to the recipient but keep some of it. This is facilitated by the creation of an extra output, which sends any remaining amount back to an address controlled by the sender, known as the "change" address (Fig. 1). This is common practice for most transactions and therefore, in most transactions, two outputs are specified, one for the recipient's address and one for the change address. It is impossible for an observer to determine which address is the recipient's and which is the change address.

A special kind of transaction output can also be specified. This is called an "OP_RETURN" output, named by the respective opcode (command) in Bitcoin's own scripting language. The inclusion of OP_RETURN outputs in the Bitcoin protocol originated from the need to have a standardized process for storing arbitrary data on the blockchain. Using an OP_RETURN output, any user that submits a transaction to the Bitcoin blockchain can store arbitrary data up to 80 bytes with each transaction. Any amount of bitcoins that are sent to an OP_RETURN output are provably unspendable, so it is also a way of "burning" bitcoins and rendering them useless (Bitcoin 2021a).

When a new transaction is broadcast to the Bitcoin network for inclusion in the next block, it is stored in what is referred to as the "mempool," a temporary storage for pending transactions. Transactions are tested for validity before being submitted to the mempool. Every time a new block is mined, the miner looks into the mempool for pending

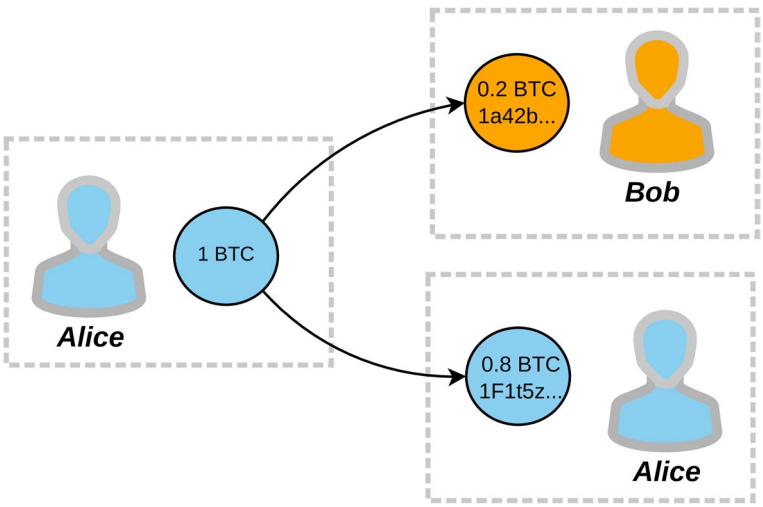

Fig. 1 A visualization of the use of a change address: Alice spends a UTXO of 1 BTC, to send 0.2 BTC to Bob (address 1a42b...), while receiving a UTXO of 0.8 BTC back as change (address 1F1t5z...)

transactions and includes them in the generated block. In case not all pending transactions can fit in the new block, as there is a limit to the overall block size and some will ultimately be left out. Transaction fees that are set aside with each transaction are moved to the miner's address with the generation of the new block. Therefore, miners have an incentive to include those transactions that offer them the most to gain. It stands to reason that if the Bitcoin network is overloaded with pending transactions, users will increase the transaction fees to incentivize miners to include their transaction into the mined block. The transaction fees are also a function of the transaction size as miners will try to fit as many transactions into the next block as these will provide them a larger amount of fees. In short, larger transactions require larger fees. The transaction size is mostly determined by the number of inputs and outputs that it has as these inflate the amount of transaction data. Ultimately, the transaction size can be taken into consideration if the fees are expressed as a function of this in satoshis/byte.

Segregated witness (SegWit) (Bitcoin 2021b) is an upgrade to the Bitcoin protocol that has been implemented, which specifies that some non-essential for all purposes parts of transaction data, called the "witness" data, and can be moved outside the transaction itself and to another, new part of the block. Among other improvements, owing to this change, SegWit transactions are smaller than previous types of transactions and their adoption has been constantly growing in the Bitcoin blockchain (Share 2021).

As the Bitcoin blockchain is public information, anyone can download and inspect it in its entirety, including all transactions that have ever happened, at any time. Therefore, all transactions can be traced back to their origins and relationships between transactions can be established (Herrera-Joancomartí 2014). This creates a need for obfuscation, as users may not want to be associated with every transaction that has occurred in their own transaction's history. One popular solution to this problem are the so called "CoinJoin" transactions (Maurer 2016). The concept was introduced in the early days of Bitcoin and has been formalized in Maxwell (2021). In these, multiple bitcoin inputs, from multiple spenders, are combined into a single transaction. This transaction has two or more outputs of exactly the same value. This way, it is impossible for outside parties to determine which spender paid which recipient or recipients. One simple way to identify such transactions is to look for transactions that have multiple (≥ 2) outputs of the exact same value. This proves to be quite effective, although the outcome may include some false positives (Maksutov et al. 2019).

Users who want to connect and interact with the Bitcoin network can do so by connecting their computer to the network, making the computer a network node. There are two major categories of nodes, full and lightweight nodes. Full nodes receive, validate, and propagate every single transaction and block throughout the network. Since that entails considerable storage requirements, as well as almost constant network activity, it is practically impossible to use them in constrained devices such as mobile phones. For such devices, lightweight nodes provide a viable alternative since they only sync transactions that are of interest to the owner. The canonical bitcoin client software is bitcoin-core (2021) and it provides a full blockchain node once installed. The downloaded blockchain data are stored in a condensed binary format on disk, which needs to be deserialized to be usable by a human. For this purpose, a Remote Procedure Call (RPC) interface is provided and can be used to access the blockchain data on disk.

Dimensionality reduction and clustering

Principal component analysis (PCA) (Dunteman 1989) is a multivariate procedure, which is commonly used for dimensionality reduction. Artificial features, known as "principal components" are created based on the original features. The principal components are calculated as linear transformations of the original features. The first few encompass most of the variation contained within the original data. Therefore, it is possible to retain only the first few principal components without losing much of the information that the original data contain. This simplification can be a very useful step for visualizing and processing high-dimensional datasets. PCA is sensitive to the scaling of the variables, so standardizing the data to a common mean and standard deviation before applying PCA is considered common practice (Jolliffe 2002).

K-means is one of the simplest and best-known unsupervised learning algorithms (Steinley 2006). It is an iterative algorithm in which the dataset is grouped into k predefined non-overlapping clusters or subgroups, making the inner points of the cluster as similar as possible. At the same time, it tries to keep the clusters as far as possible. Data points are allocated to a cluster if the sum of the squared distance between the cluster's centroid and the data points is at a minimum. Although it is a relatively simple algorithm, it has been known to perform well, especially with respect to financial data (Kou et al. 2014). The cluster's centroid is the arithmetic mean of the data points in the cluster. Perhaps the most important decision when applying the k-means algorithm is the selection of the value of k that will be used. A popular method of doing this is employing the so-called "elbow method." The idea behind this method is to run k-means clustering on the dataset for a range of values of k, calculate the sum of squared errors (SSE) for each value of k. If the SSE values are plotted against the respective values of k, the plot usually resembles an arm. The optimal value of k is then at the point that the elbow of the arm is located. Other methods for determining the optimal number of clusters have been suggested in literature, such as the use of the F-statistic method when using fuzzy cluster analysis (Li et al. 2022). The trimmed k-means algorithm (Cuesta-Albertos et al. 1997) is a variant of the algorithm that produces better clustering results for noisy data as these are removed prior to the applications of the algorithm. The proportion of noisy data to be removed can be adjusted. This can be achieved by setting the desired value of a, which represents the proportion of data points that should be trimmed. It takes values within the range [0, 1] and its value is typically determined by trial-and-error.

Related work

This section briefly reviews related work in the literature.

One of the first works discussing user privacy and unveiling their profiles based on their transactions is Androulaki et al. (2013). This study relies on a simulator that mimics the use of Bitcoin in a realistic, for the time, university setting. Using behavior-based clustering techniques, they manage to identify 40% of user profiles that were involved in the study. Another early study (Lischke and Fabian 2016), manages to identify a gambling network that features many very small transactions by clustering bitcoin addresses in a user graph. Maesa et al. (2018) also created a user graph derived from the transactions data stored in the Bitcoin blockchain. A total of three major clusters of transactions are identified, where exchanges, miners, and users that store bitcoins with little spending

are assigned. Maesa et al. (2016) analyzed approximately 100 million Bitcoin transactions, and then create a user graph to analyze user behavior. They discovered the existence of central nodes acting as privileged bridges between different parts of the network.

In Meiklejohn et al. (2013), the authors engaged in various kinds of activities on the Bitcoin network, including mining, using online wallet services, exchanges, purchasing goods, gambling, as well as bitcoin laundry services to establish a ground truth for some transactions at least. They applied address clustering heuristics to identify illicit activity by addresses belonging to the same user.

Zhang et al. (2020) propose a novel heuristic for clustering addresses in the Bitcoin blockchain. The authors argue that this makes a considerable improvement in effectively identifying addresses that are controlled by the same user. However, in the absence of ground-truth labels, it is very difficult to assess the quality of the proposed clustering heuristic.

Wu et al. (2020) propose a Bitcoin transaction network analytic method for facilitating Blockchain forensic investigation based on an extended safe Petri Net. This takes transaction structure and behavior features into account, in addition to address clustering and Bitcoin flow analysis methods. A total of 19 features are identified to define Bitcoin transaction patterns for analyzing and finding suspected addresses. The efficiency of the proposed method is empirically verified based on a real-life case study analysis; however, this concerns an event that occurred several years ago when the Bitcoin network and transactions had different properties than they do today. Harrigan and Fretter (2016) present the primary reasons behind the effectiveness of Bitcoin address clustering. These include address reuse, avoidable merging, and super-clusters with high centrality, as well as the incremental growth of address clusters. However, these also refer to transaction properties assumptions that are probably not true anymore.

In a different approach, Zola et al. (2022) employ three different deep learning techniques along with unsupervised anomaly detection methods to perform network traffic analysis through node behavior classification.

Caprolu et al. (2021) provide evidence that clustering techniques that rely on addresses for deanonymization purposes can no longer be efficiently applied. According to the authors, this is due to the recently increased number of non-standard transactions that occur in the Bitcoin blockchain. The authors introduce a framework for parsing such non-standard transactions and study and classify them according to emerging patterns, with promising results.

Hirshman (2013) use a clustering algorithm, k-means, and a role detection algorithm, RolX, to reveal anomalies in the bitcoin transaction network. Data were split in five clusters, with one having properties that pointed to anomalous behavior that may indicate illicit activity. The authors state that there is no way of proving their findings owing to the lack of labeled data. This work also refers to the first few years of activity in the Bitcoin network, when its properties were different than they are today.

More recently, Nerurkar et al. (2020) present a supervised learning model for identifying illegal activities in Bitcoin. To evaluate the model, a dataset of 1216 entities on Bitcoin was extracted from the Blockchain. The model used nine features as input and was trained for segregating 16 different licit-illicit categories of users. The authors conclude that the proposed model provided a reliable tool for forensic study.

Weber et al. (2019) contribute a dataset consisting of a time series graph of over 200K labeled Bitcoin transactions. In an effort to categorize transactions as illicit or not, they perform several binary classification techniques including variations of logistic regression, random forest, multilayer perceptrons, and graph convolutional networks analyses. Results indicate that random forest is the superior algorithm for such purpose. A prototype for visualization of transaction data is also provided.

Nerurkar et al. (2021) analyze the Bitcoin network to understand how the social and anti-social tendencies in the user base affect its evolution. A total of 20 types of legal and anti-social entities operating on Bitcoin are uncovered and a methodology for doing so based on network properties is provided, using a supervised learning technique for identifying unknown wallets.

Harlev et al. (2018) use supervised machine learning to predict the type of yet-unidentified entities. This work takes advantage of identities of entities that have been revealed and uses them as training data to build classifiers. The authors find that they can predict the type of entities not yet identified with good accuracy.

Jourdan et al. (2018) examine the information revealed by the pattern of transactions in the vicinity of an entity transaction. This is used to characterize entities in the Bitcoin network. The authors use a graph network and novel features to classify entities according to their behavior. They find that an attacker can identify an entity type with relative accuracy using a very simple model of logistic regression, but with a carefully selected set of features.

Monamo et al. (2016) examine the effectiveness of the k-means clustering algorithm to detect fraudulent activity in Bitcoin transactions. Both the "classical" k-means algorithm and a trimmed k-means alternative are employed to this end. Results reveal some disparities between the two clustering algorithms in the presence of outliers. Owing to its sensitivity to anomalies, K-means formed a spurious cluster containing very few objects. In contrast, using the trimmed k-means algorithm, the spurious cluster attained from k-means is filtered out and improvements in group structures are realized as a result. This work also relies on grouping addresses, based on heuristics, prior to applying the clustering algorithms, to aggregate data and create some of the features that are extracted.

To detect suspicious users and transactions in the Bitcoin network, Pham and Lee (2016) use unsupervised learning methods. The authors report that they are able to detect two known cases of theft and one known case of lost funds, although their evaluation metrics are not high. Bartoletti et al. (2018) collected Bitcoin addresses that are known to be related to Ponzi schemes. Using these, they evaluated various supervised machine learning algorithms to identify the Ponzi schemes with only 1% of false positives. Chen and Tsourakakis (2022) use anomaly detection through a novel technique that finds subgraphs in financial networks to identify abnormal behavior in another blockchain network, Ethereum. Prado-Romero et al. (2018) use anomaly detection techniques to discover bitcoin mixing services and the addresses associated with them. The authors find that their approach is effective by comparing their results to accounts that are known to be controlled by mixing services. Shayegan et al. (2022) employed a trimmed k-means algorithm to perform collective anomaly detection on users of the Bitcoin network, revealing fraudulent users by analyzing their behavior. Sayadi et al.

(2019) propose a new method for detecting anomalies in bitcoin transactions using One Class Support Vector Machines and k-means clustering algorithms in two stages. Shafiq (2019) examine various anomaly detection techniques on transactional network data in Bitcoin, focusing on identifying anomalous patterns using unsupervised learning techniques. The results seem promising, but the lack of more publicly available data affects the quality of anomaly detection.

As is evident, most studies rely on external information to provide labels that will act as the ground truth for training supervised algorithm models and evaluating results. It is also the case, that most works try to identify and cluster addresses based on heuristics, instead of transactions. The absence of a significant volume of trustworthy labeled data is an inherent difficulty in evaluating the performance of unsupervised algorithms.

Our work differentiates from previous studies by applying an unsupervised learning method to cluster transactions, instead of addresses or users. To this end, we introduce some novel features not previously found in literature. While most previous studies focus on binary classification for detecting fraudulent activity, using these novel features, along with our data processing pipeline, it is possible to classify transactions in several clusters according to different patterns.

Methodology

In this section, we provide an outline of the Bitcoin blockchain data how the features used for this study were extracted. A description of these features follows while the section concludes with a summary of the machine learning clustering techniques that were adopted.

The data processing pipeline for Bitcoin blockchain transactions is displayed in Fig. 2 and presented in detail as follows.

Bitcoin blocks from block 610,000 (mined December 27, 2019) to block 660,000 (mined December 5, 2020), covering almost a year of transactions in the Bitcoin network, were considered. The total number of transactions included in these blocks amounted to 105,589,345. This means that an average number of more than 300,000 transactions are submitted to the Bitcoin blockchain on a daily basis. Since we wanted our results to be representative of the latest advancements and trends in the Bitcoin blockchain, it was

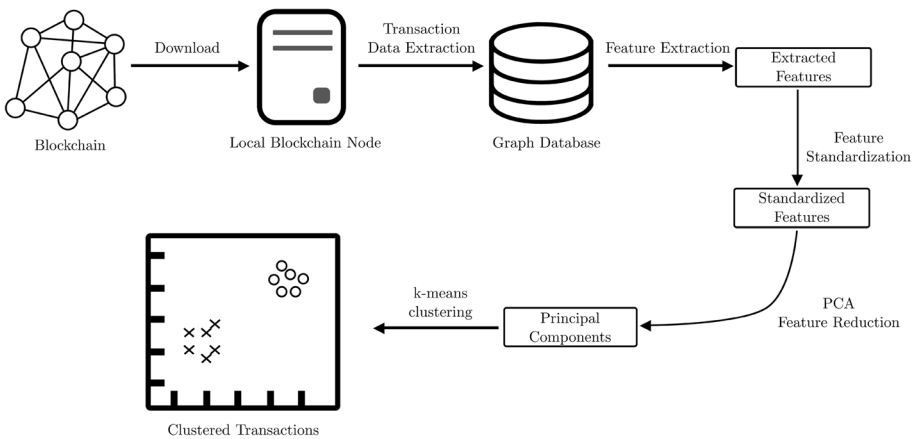

Fig. 2 Data processing pipeline for Bitcoin blockchain transaction data

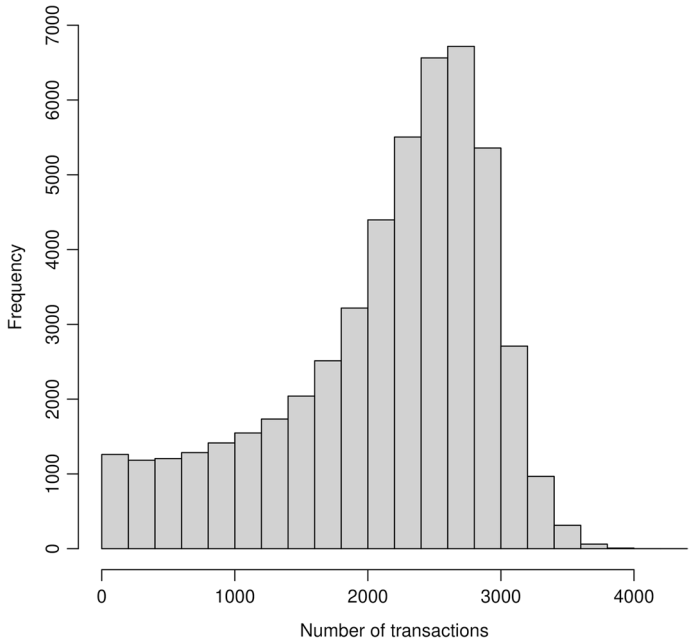

Fig. 3 Distribution of the number of transactions for Bitcoin blocks 610,000 to 660,000

Table 1 Descriptive statistics for transactions/block

Min.	Q1	Median	Mean	Q3	Max.	SD
1	1672	2323	2112	2708	4377	804.74

necessary to restrict the number of blocks to include going back in time. Older blocks may exhibit different usage patterns that could possibly taint our results. Figure 3 presents the distribution of the number of transactions per block for that block range, while Table 1 provides some basic statistics for the same data. It is noteworthy that numerous blocks that include only just one transaction (the Coinbase transaction) exist. On the other hand, for blocks that include a large number of transactions, the majority of these transactions have to be SegWit transactions, which are generally much smaller in size. After a closer examination of these blocks, it was confirmed that most are indeed SegWit transactions.

Data model and preparation

For the purpose of downloading the Bitcoin blockchain data, we connected a node to the Bitcoin network and set it to synchronize and index all the transactions that are included in the blockchain. The node was running the bitcoin-core client, version 0.20.0. The blockchain transaction data along with the respective indices occupied more than 380 GB of storage space. To easily access the data, we enabled the bitcoin daemon's RPC interface. Data from the entire blockchain starting from the genesis block up until block 660,000 was then entered into a Neo4j graph database (Neo4j 2021). Even though our goal was to examine transactions in recent blocks only, the inclusion of the entire blockchain was necessary to calculate the values for some of

the novel features we considered. Our goal was to create a tree of transactions that can be easily traversed and queried using Neo4j's own query language.

We model the data as a graph $\mathcal{G}$ with nodes as $\mathcal{N}$ and edges $\mathcal{E}$.

- *Nodes* There are multiple node types for blocks, transactions, transaction outputs, and addresses. We denote the set of these nodes as $\mathcal{N}_b$, $\mathcal{N}_t$, $\mathcal{N}_o$, and $\mathcal{N}_a$, respectively, and it holds that $\mathcal{N} = \mathcal{N}_b \cup \mathcal{N}_t \cup \mathcal{N}_o \cup \mathcal{N}_a$.
- *Edges* in the graph are directed and represent the relationships between nodes. The relationships have a different meaning depending on the types of nodes the edges connect. In the following, we describe what types of edges exist in the graph and their meaning. For ease of reference, this information is summarized in Table 2 as well.

  - *Edges between blocks*, $\mathcal{E}_{bb}$ Each block node has an outgoing edge to another block node, with the exception of the genesis block, which has no incoming edge. The block nodes are connected based on chronological order from oldest to newest.
  - *Edges between blocks and transactions*, $\mathcal{E}_{bt}$ A block includes one or more transactions. In the graph G, we denote this relationship with a directed edge from the block to the corresponding transaction.
  - *Edges between transactions and transaction outputs*, $\mathcal{E}_{to}$ A transaction can create one or more transaction outputs.
  - *Edges between transaction outputs and transactions*, $\mathcal{E}_{ot}$ One or more outputs are spent by a transaction with the exception of Coinbase transactions, which do not spend any outputs.
  - *Edges between transaction outputs and addresses*, $\mathcal{E}_{oa}$ A transaction output may be locked to an address, and this is represented by an edge from a transaction output node to the corresponding address node.

  It holds that $\mathcal{E} = \mathcal{E}_{bb} \cup \mathcal{E}_{bt} \cup \mathcal{E}_{to} \cup \mathcal{E}_{ot} \cup \mathcal{E}_{oa}$. The resulting graph is a *directed acyclic graph* (DAG).

A partial example of this graph is illustrated in Fig. 4. Block number N is followed by block number $N+1$, which is in turn followed by block number $N+2$. Block N includes (among others) transaction *270d7*. Transaction *270d7* spends two transaction outputs, which hold a value of *1.5 bitcoins* and *3.5 bitcoins*, respectively. The transaction output that holds a value of *3.5 bitcoins* is shown to be locked to address *3wR74*. Transaction *270d7* spends these two transaction outputs and in turn creates a single transaction output, which holds a value of *5 bitcoins*. The sum of all outputs any transaction spends is always equal to that it creates, in bitcoins. The transaction output that holds a value of *5 bitcoins* is locked to address *15yWB*. This transaction output is spent by transaction *c17d8* (included in block $N+1$), which creates two new transaction outputs with values of *3 bitcoins* and *2 bitcoins*, respectively. The transaction output that holds a value of *3 bitcoins* is shown to be locked to address *1aZdk*. Transaction *35,064* (included in block $N+2$) spends this output, along with two

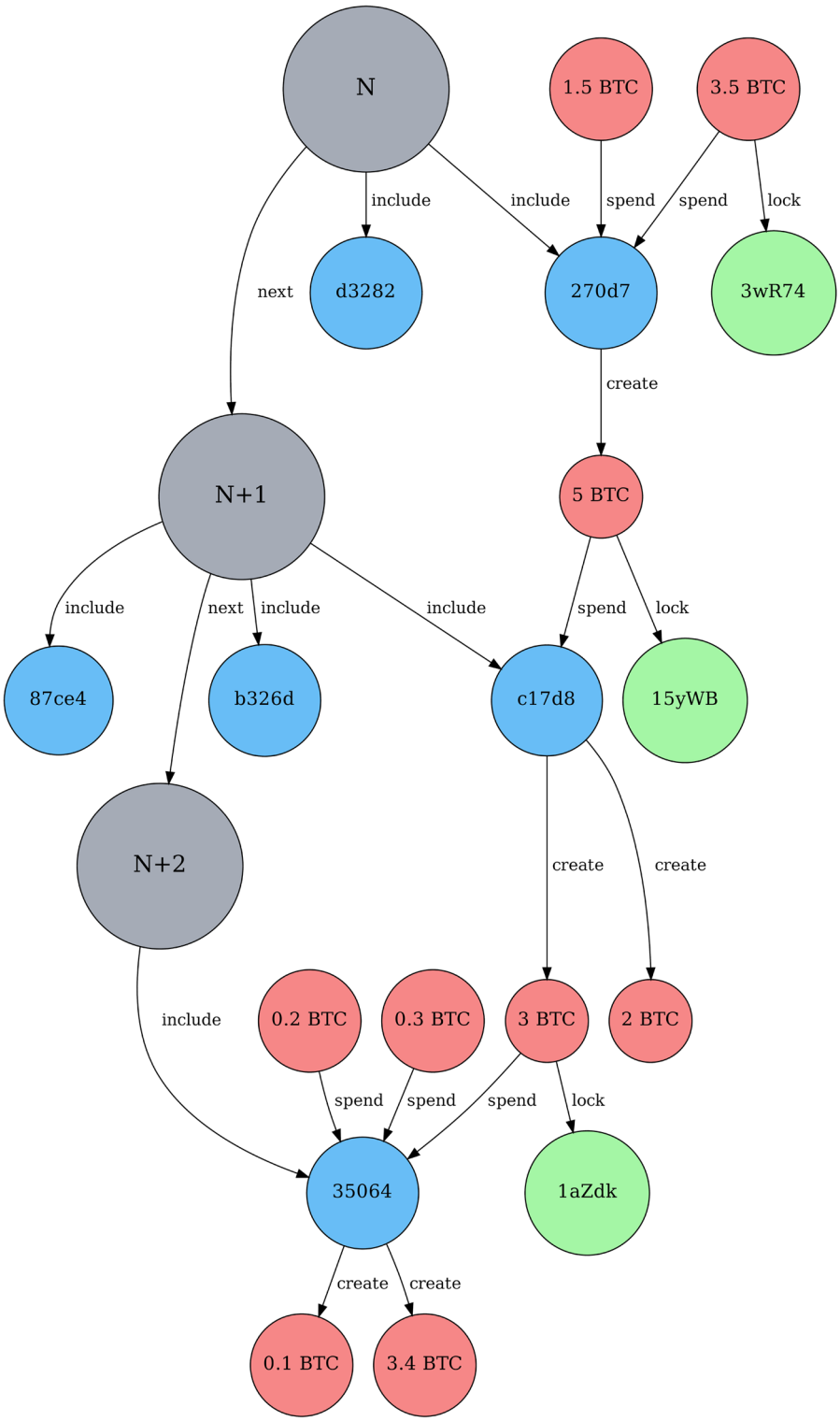

Fig. 4 Example of a partial graph as stored in the graph database (: block, : transaction, : UTXO, : address)

Table 2 Types of edges in the graph $\mathcal{G}$

Notation	From node type	To node type	Meaning
$\mathcal{E}_{bb}$	Block	Block	Chronological order
$\mathcal{E}_{bt}$	Block	Transaction	A block includes transactions
$\mathcal{E}_{to}$	Transaction	Transaction output	A transaction creates transaction outputs
$\mathcal{E}_{ot}$	Transaction output	Transaction	A transaction output is spent by a transaction
$\mathcal{E}_{oa}$	Transaction output	Address	A transaction output is locked to an address

more outputs that hold values of *0.2 bitcoins* and *0.3 bitcoins*, to create two more outputs, which hold values of *0.1 bitcoins* and *3.4 bitcoins*, respectively. The outputs that hold values of *2 bitcoins*, *0.1 bitcoins*, and *3.4 bitcoins* remain unspent in this example. For the sake of simplicity, any transaction fees that the transactions may have incurred are not shown in this example. It should also be noted that transaction outputs are indexed in our database using their respective hashes but are presented here using their bitcoins values, again, for the sake of simplicity. All UTXOs are locked to an address, so all UTXO nodes always have an outgoing edge to an address node. However, to reduce clutter in this partial example, this relationship is shown only for some of the UTXO nodes in Fig. 4.

The resulting Neo4j database occupied more than 1 TB of disk space. The scripts that were used to transform the raw blockchain data to the Neo4j database have been uploaded and are available from our source code repository at Gitlab (Project 2021).

Feature engineering

For each transaction, the following features were extracted:

Number of inputs: the number of previously unspent transaction outputs used as inputs for the transaction. For Coinbase transactions, this was set to zero as such transactions are the exception to the rule that all transactions must have at least one input.

Number of outputs: the number of outputs that the transaction created. Any OP_RETURN outputs that may have been specified have been included in this count.

Total amount: the amount of bitcoins that were transferred with the transaction, including transaction fees. With the exception of Coinbase transactions, where there are no inputs, the total amount specified as inputs to a transaction always match the total amount specified as outputs of the same transaction.

OP_RETURN count: the number of OP_RETURN outputs of a transaction. In the vast majority of transactions, this number is equal to zero, or one. Only a single OP_RETURN output is allowed by the Bitcoin protocol. However, there are some rare exceptions (Bistarelli et al. 2019), with transactions including more than one OP_RETURN outputs. The Coinbase transaction of block 650,000, which includes three OP_RETURN outputs, is such an example. We

considered this as a feature as OP_RETURN outputs are the most common way to store arbitrary data on the blockchain. As such, they are likely used more often by applications that utilize the immutability of the blockchain to store data. These could include identity verification data or data written for notarization purposes. While OP_RETURN outputs are the cheapest way to store arbitrary data on the blockchain, their use still incurs significant fees. Therefore, they are unlikely to be used when users just want to send some bitcoins.

Fees: the fees in satoshis/byte that were spent for including the transaction in the blockchain. These are calculated by taking the amount paid to the miners as fees for the transaction divided by the total size of the transaction in bytes. For SegWit transactions, the "witness" data of a transaction, which includes the signature, has been moved out of the transaction data into a separate part of the Bitcoin block and are not taken into account.

Coinbase distance: all bitcoins that are circulated in the Bitcoin blockchain with transactions stem from their creation in a Coinbase transaction. Therefore, it is possible to trace the origin of any transaction input back in time to one or more Coinbase transactions. This is done by traversing the transaction graph using a breadth-first-search algorithm. We set the Coinbase distance for each transaction as the number of hops in the transaction graph to traverse back in time to reach the most recent Coinbase transaction. Our rationale for including this feature was that miners possibly spend their earnings as soon as they can, to send their bitcoins to more secure wallets, or even to online cryptocurrency exchanges where they can trade their bitcoins for fiat currency. Therefore, it could potentially work as a significant factor for clustering transactions related to mining activities.

CoinJoin distance: we identified all the transactions in the Bitcoin blockchain that have the characteristics of a CoinJoin transaction (two or more outputs with the exact bitcoin amount specified). For every transaction in our dataset, we calculated its distance in hops in the transaction graph from the closest transaction that was identified as such, searching into past blocks as well as blocks produced after the transaction was included in the blockchain. An example of such a transaction, which appears in block 650,000, is presented in Fig. 5. If there was no transaction that was identified as a CoinJoin one, the CoinJoin distance was set to 660,000, which was the maximum block number in our study. The idea behind considering this as a feature is that CoinJoin transactions are probably mostly used to mix bitcoins obtained by illicit means as it makes tracking those bitcoins much harder (Han et al. 2020). Therefore, low values may indicate transactions

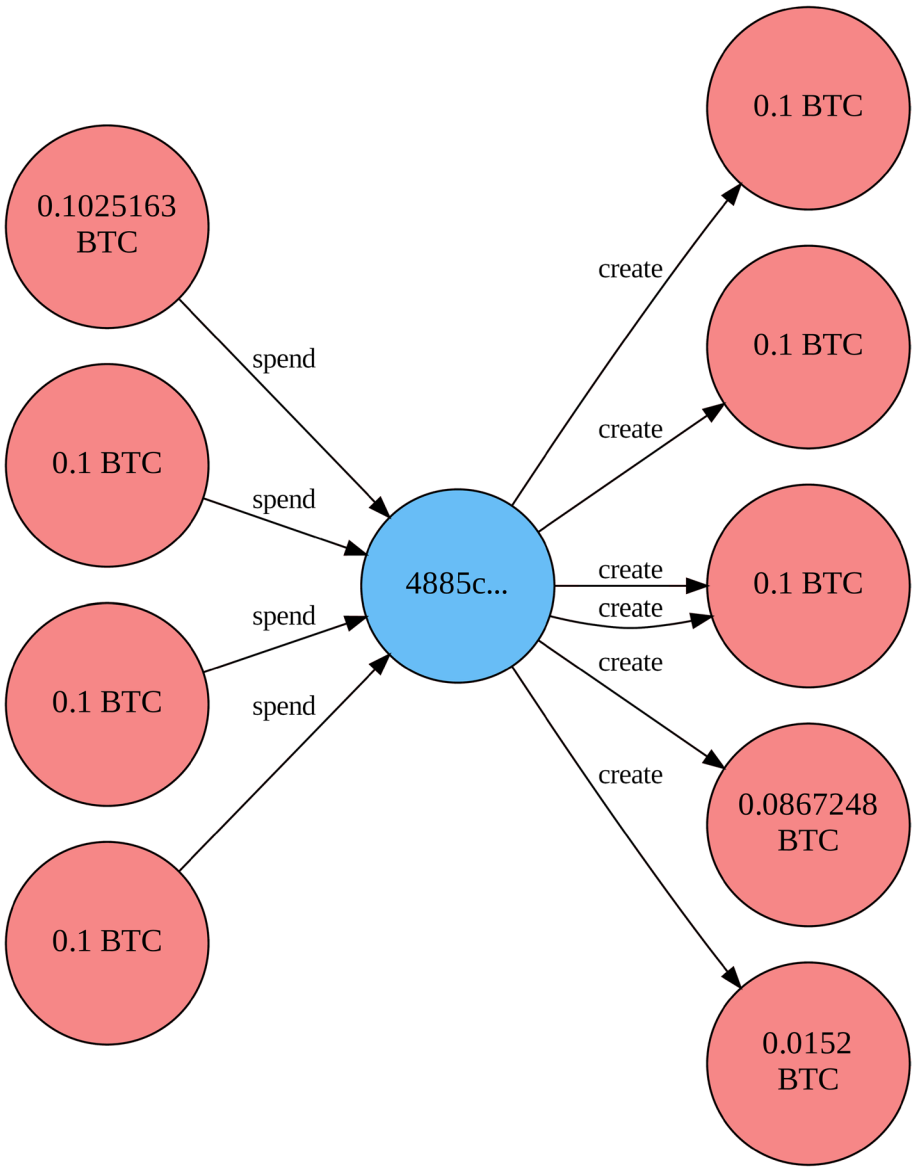

Fig. 5 An example of what seems like a CoinJoin transaction, found in block 650,000

that are more likely to have been obtained illegally. Examples may include bitcoins obtained from ransomware attacks or drug trafficking.

Blocks unspent: for each transaction output we counted the number of blocks since the one that included the transaction which created the output until the block that included the transaction which spent the output. For transactions that had not been spent until block 660,000, this value was set to 660,000, the maximum block distance within our dataset. The average value for all outputs was used for this feature.

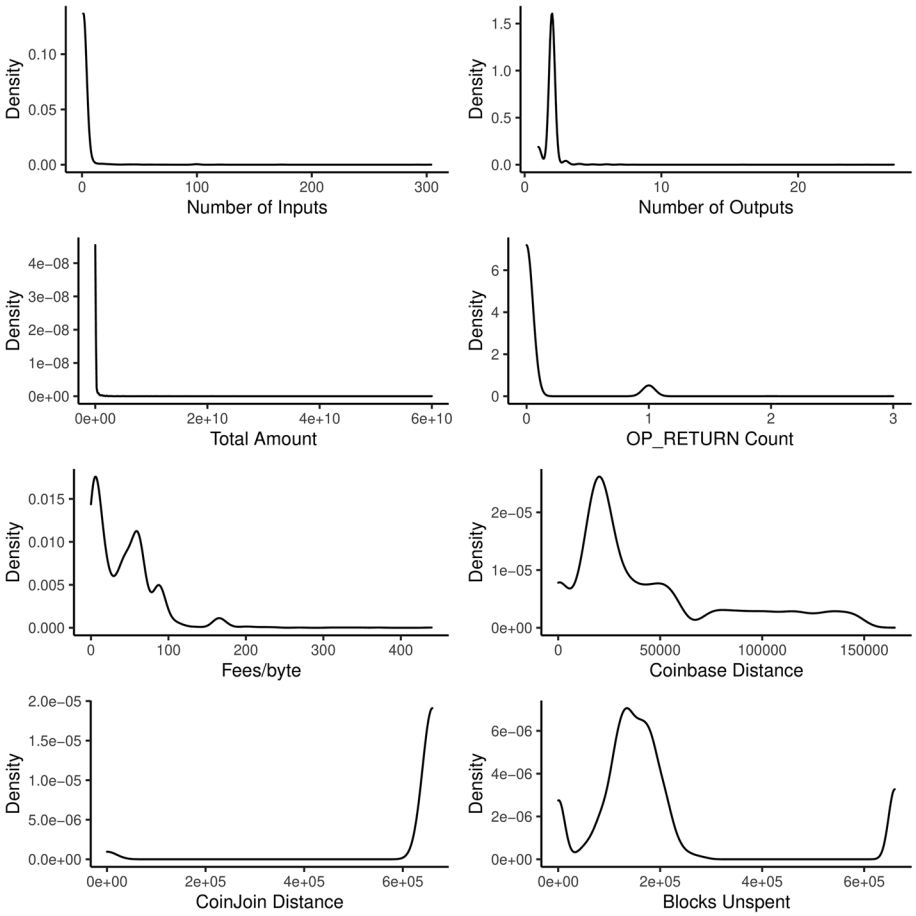

Fig. 6 Density plots for initial features

Table 3 Correlation matrix for initial features

	Inputs	Outputs	Amount	OP_RETURN	Fees	Coinbase dist.	CoinJoin dist.
Outputs	− 0.048						
	(0.033)						
Amount	− 0.009	0.016					
	(0.700)	(0.471)					
OP_RETURN	− 0.016	− 0.001	− 0.020				
	(0.484)	(0.954)	(0.360)				
Fees	− 0.057	− 0.026	0.097	0.051			
	(0.011)	(0.240)	(0.009)	(0.023)			
Coinbase dist.	0.018	0.025	− 0.027	0.015	− 0.032		
	(0.469)	(0.307)	(0.291)	(0.511)	(0.187)		
CoinJoin dist.	0.015	0.009	− 0.002	− 0.007	0.003	− 0.004	
	(0.484)	(0.698)	(0.653)	(0.752)	(0.615)	(0.598)	
Blocks Unspent	0.008	− 0.005	0.004	− 0.012	0.012	0.021	− 0.007
	(0.699)	(0.813)	(0.858)	(0.512)	(0.520)	(0.357)	(0.714)

Pearson r, *p* values in parentheses

To the best of our knowledge, the Coinbase distance, CoinJoin distance, and blocks unspent features have not been used in literature before.

The distribution of data in the features examined is illustrated in Fig. 6. Table 3 presents a correlation matrix between features. Only a few correlations are statistically significant. There is a negative correlation between the number of inputs and outputs. This probably comes from the fact that most of the transactions that have a large number of inputs are consolidation transactions, which gather several UTXOs into one. There is also a significant negative correlation between the number of inputs and the fees per byte, which demonstrates the fact that while the total amount of fees may increase when having more inputs, it is more cost-effective to combine several transactions into one, something that cryptocurrency exchanges are known to do. There is a positive correlation between the total amount transacted and the fees per byte spent. This may reflect that when dealing with larger amounts, people are willing to spend more on fees, so their transaction is more likely to be included in the next block that will be mined. Finally, there is a positive correlation between the OP_RETURN count feature and the fees per byte. In this case too, it seems that people that want to use the blockchain for storing data are willing to spend more on fees than average.

Transaction clustering pipeline

The extracted features were then processed as follows:

1 *Data standardization* All features were standardized to a mean value of 0 and a standard deviation of 1 prior to performing a PCA. Standardization is a typical step performed prior to PCA and is used so that each of the initial variables contributes equally to the analysis (Jolliffe 2002). The reason this is important is because PCA is very sensitive with respect to the variances of the initial variables. If there are large differences between the variances of the initial variables, those with larger ranges will dominate over those with small ranges; this will lead to biased results. Transforming the data to comparable scales by standardizing them can prevent this problem.

2 *Dimensionality Reduction* PCA was carried out to reduce the dimensionality of the data, assisting the clustering algorithm that will be performed later and making clustering calculations faster (Ding and He 2004). Clustering algorithms such as k-means have difficulty accurately clustering data of high dimensionality. Our dataset is not necessarily highly dimensional as it contains eight features, but even this amount may create issues for k-means. We adopted the typical procedure of utilizing a scree plot with the elbow method for visually determining the optimal number of components to retain.

3 *Clustering* A k-means clustering algorithm was applied to the most important principal components as identified by the PCA. In particular, a trimmed k-means clustering method (Cuesta-Albertos et al. 1997) was applied so that outliers that could interfere with the analysis would be removed before clustering. The elbow method was employed in this case as well to determine optimal number of clusters. The proportion of outliers to be removed by the algorithm a was determined via a trial-and-error process for several values of a and the visual inspection of the respective results. The resulting clusters of data points were plotted against the two most

important principal components and an attempt was made to summarize the cluster characteristics using domain specific knowledge.

4 *Cluster Validation* Finally, the Silhouette index was calculated to assess the cluster cohesion and validate the results of the clustering algorithm (Rousseeuw 1987). The silhouette index receives values between −1 and 1, with values close to 1 indicating that results are well-clustered, values around 0 indicating that cases may be misplaced between clusters, and values close to −1 indicating that cases are misclassified. As quality measurement of a clustering algorithm has shown to be as important as the clustering algorithm itself (Arbelaitz et al. 2013), it is important to assess the cluster cohesion based on an objective evaluation measure. In addition, we evaluated our results against labeled data acquired from a public database of addresses that are known to be controlled by online exchanges, mining pools, lending, and gambling services, among others.

All analyses were carried out using GNU R 4.0.3 (R Core Team 2020).

Experimentation

This section presents the results of the analyses described in the "Methodology" section, and provides some discussion on the findings.

Following standardization of the original features, a PCA was performed. A scree plot was used to determine the number of components to retain (Fig. 7). By inspecting the scree plot, we concluded that three or four principal components were adequate for describing our dataset without losing much information. Indeed, the variance of information contained within these components was 88.7% and 91.3%, respectively. We finally decided on retaining three principal components as the additional information provided by the fourth was relatively small, taking the additional complexity that keeping it would entail into account.

The structure of all principal components that resulted from the PCA is presented in Table 4. The table includes the coefficients for the principal components that PCA

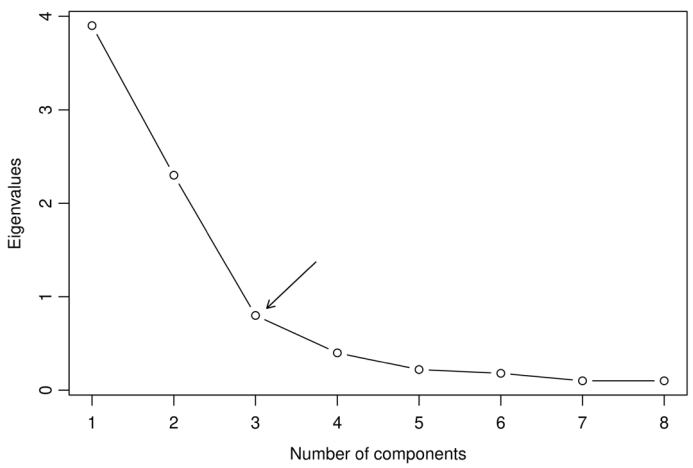

Fig. 7 Scree plot for determining the number of components to retain in PCA

Table 4 Principal components as linear combinations of the initial variables

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8
Number of inputs	0.459	− 0.145	0.030	− 0.676	− 0.347	− 0.241	0.494	0.018
Number of outputs	0.484	− 0.135	− 0.057	− 0.401	0.240	0.622	− 0.357	0.103
Total amount	0.466	− 0.277	0.052	− 0.004	− 0.212	− 0.175	− 0.487	− 0.657
OP_RETURN count	− 0.067	0.219	− 0.739	0.091	0.116	− 0.035	− 0.085	0.487
Fees	− 0.314	0.444	0.017	0.122	− 0.017	− 0.014	− 0.023	0.368
Coinbase distance	− 0.237	0.585	− 0.075	0.366	0.436	0.143	0.568	− 0.348
CoinJoin distance	− 0.123	0.452	− 0.662	− 0.078	− 0.281	0.681	0.245	− 0.196
Blocks unspent	− 0.404	0.304	0.058	− 0.468	0.703	− 0.195	− 0.022	− 0.158

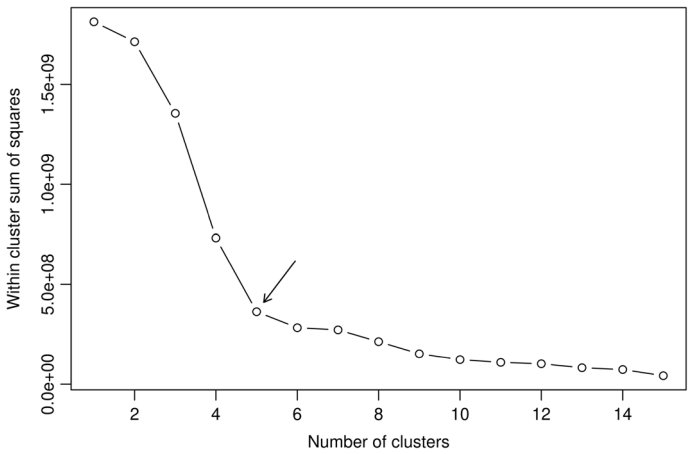

Fig. 8 Elbow plot for determining the optimal number of clusters

has calculated with respect to the initial variables. As an example, if we name our initial variables X_1 through X_8, then the first principal component Z_1 is calculated as:

$$Z_1 = 0.459X_1 + 0.484X_2 + 0.466X_3 - 0.067X_4 - 0.314X_5 - 0.237X_6 - 0.123X_7 - 0.404X_8$$

Higher absolute values of coefficients indicate that the respective initial variable is more significant in contributing to the value of the principal component. By inspecting the results, we can determine that PC1 primarily represents the number of inputs and outputs, the total amount spent and the number of blocks that the outputs of the transaction have stayed unspent for. Additionally, PC2 primarily represents the distance from a Coinbase transaction, the distance from a CoinJoin transaction, as well as the transaction fees, while PC3 is associated with the number of OP_RETURN outputs in the transaction and the distance from a CoinJoin transaction.

The k-means clustering algorithm was iterated for $1 \leq k \leq 15$ on the first three principal components. The optimal number of clusters k was decided using the within clusters sum of squares metric, applying the well-established elbow method. While the number of clusters could be infinitely large, we determined that there was no reason to try values of $k > 15$ for any practical purposes. As documented in Fig. 8, the optimal number of clusters seems to be five, where the elbow of the curve is formed.

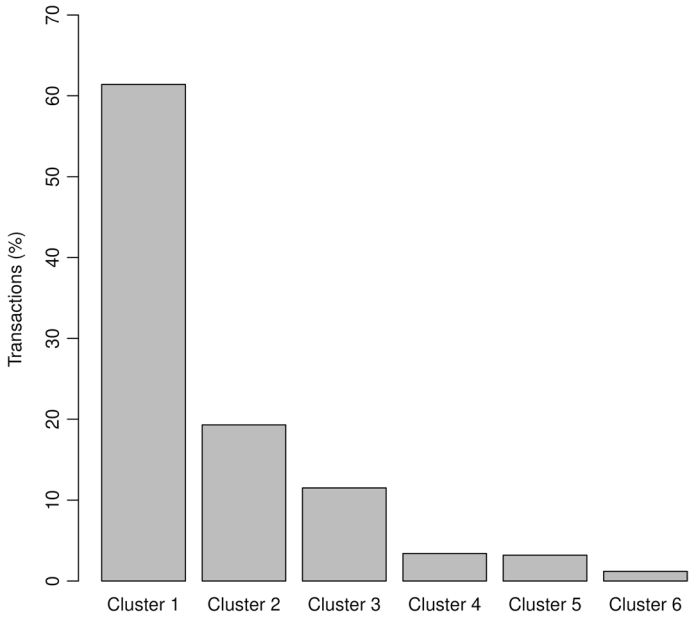

Fig. 9 Distribution of k-means clustering results

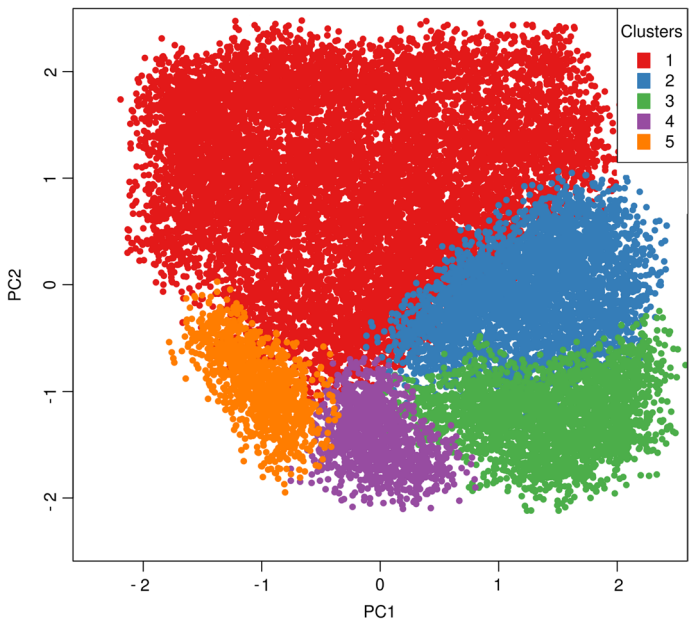

Fig. 10 Clustering results on the two first principal components

The proportion of outlier data points that will be removed prior to performing the k-means algorithm was set to $a = 0.01$. This decision was based on the visual inspection of 2-dimensional plots of the clustered data against the largest three principal components, similar to Fig. 10 for $0.005 \leq a \leq 0.05$ in steps of 0.005. The objective was to remove any data points that were obvious outliers, without compromising the clustering process by removing excess data.

The distributions of transactions in the clusters that were formed by the k-means algorithm with $k = 5$ are illustrated in Fig. 9. An additional cluster (No. 6) has been added. This includes all transactions that were trimmed out of the k-means clustering algorithm as outliers. The latter only constitutes about 1% of total transactions as the previously set value of a would indicate and the data points assigned to it do not actually form a cluster as such; rather, it is a collection of random points. The largest cluster is the first cluster, including 61.4% of transactions. The second cluster includes 19.3% of transactions, while the third includes 11.5%. The remaining two clusters constitute a little more than 3% of the total number of transactions each.

Figure 10 displays a 2-dimensional plot, including the five identified clusters as represented by various color codes, using the two largest principal components as plot axes. Transactions removed as outliers by the k-means algorithm are not shown so as to provide a better visual representation of the five primary clusters. It is also apparent that the first cluster is the one with the most transactions classified in it.

The Silhouette index value was calculated for $k = 5$ clusters and found to be equal to 0.78, indicating that most transactions are well-matched to the clusters they have been assigned to, although the match may not be perfect.

To try to summarize the clusters according to the transactions assigned into each of them, the mean value and standard deviation for each of the original features and for each cluster were calculated, which are reported in Table 5.

Cluster 1: the most prominent cluster includes transactions with the lowest count of inputs and outputs. The total transaction amount is also the smallest among all clusters. The transaction fees are not too high, but not too low either. All these provide a strong indication that these are most probably common user transactions, moving bitcoins between wallets. Payments from users

Table 5 Cluster summary [mean (± SD)]

	Cluster 1	**Cluster 2**	**Cluster 3**
Number of inputs	1.3 (±2.51)	10.3 (±9.34)	2.2 (±3.47)
Number of outputs	2.1 (±3.13)	14.4 (±13.38)	4.3 (±3.32)
Total amount	0.8 (±3.21)	32.9 (±14.21)	12.7 (±28.3)
OP_RETURN count	0.031 (±0.186)	0.003 (±0.421)	0.008 (±0.136)
Fees	102.3 (±64.32)	28.4 (±15.87)	30.9 (±24.32)
Coinbase distance	9139.3 (±14328.13)	6232.6 (±8932.12)	105.3 (±31.54)
CoinJoin distance	12291.1 (±13178.47)	10988.3 (±19212.18)	8821.3 (±10219.91)
Blocks unspent	24034.7 (±29021.22)	134.4 (±110.35)	15.6 (±35.19)
	Cluster 4	**Cluster 5**	**Cluster 6**
Number of inputs	3.4 (±4.49)	1.6 (±2.29)	1.6 (±3.64)
Number of outputs	2.8 (±3.41)	2.7 (±4.48)	2.9 (±2.89)
Total amount	5.6 (±9.31)	2.6 (±5.87)	3.6 (±14.21)
OP_RETURN count	0.001 (±0.382)	0.062 (±0.542)	0.034 (±0.211)
Fees	167.6 (±73.38)	243.5 (±80.74)	154.2 (±61.35)
Coinbase distance	8782.1 (±12128.26)	7658.2 (±19217.86)	9554.2 (±7812.22)
CoinJoin distance	129.2 (±324.21)	9298.7 (±9826.43)	12871.6 (±16129.46)
Blocks unspent	9.8 (±21.91)	28755.5 (±41288.11)	31856.8 (±72132.83)

to merchant services may be included in this cluster. Indeed, most transactions originating from common Bitcoin wallet applications try to restrict the number of inputs to the minimum, which is one input, to keep transaction fees low. In the case that is not possible, they will then try to combine two UTXOs to provide the bitcoin amount that is needed for the transaction. Only in the case that this also fails, they will try to combine three or more UTXOs for the same purpose. With respect to the number of outputs for each transaction, the most common practice with Bitcoin wallets is to create an additional output sending any remainder amount of an input that the sender does not want to send to the recipient back to a change address, which is usually generated for such purpose and that transaction alone. Most common users only want to send some amount to a single recipient with each transaction. In fact, most wallet applications do not allow specifying multiple recipients for a transaction, or make it cumbersome to do so. Hence, the most common number of outputs for common user transactions would be to have a single output for the recipient and another one for the change, for a total of two outputs. This goes well with the fact that most transactions included in Bitcoin have a single input and two outputs, as is evident to anyone that browses the bitcoin blockchain with any online blockchain explorer software. Furthermore, it stands to reason that most common users would only transfer moderate amounts of bitcoin with their transactions, something that reinforces the assumption that transactions in this cluster are originated by common users.

Cluster 2: this cluster is characterized by the highest number of inputs and outputs as well as the highest average for the total amount transacted. On the other hand, the transaction fees are the lowest among all clusters. The number of blocks that outputs remain unspent is also relatively low, although not the lowest. These fit well with the way that high volume actors, such as online exchanges, and lending services function in the Bitcoin blockchain. These try to keep fees as low as possible by aggregating multiple inputs and multiple outputs into a single transaction. For example, if 10 people withdraw bitcoins around the same time, the exchange will not create one transaction for each recipient, but would rather create a single transaction with 10 outputs, one for every recipient. This single transaction will occupy less space in the blockchain compared to the space that 10 separate transactions would have occupied, resulting to lower fees in total. It is also a common practice among Bitcoin users to move their bitcoins to different addresses, stored in hardware wallets or paper wallets, not long after they receive their bitcoins from an exchange. This is especially true for higher amounts. This cluster may also represent transactions related to the lightning network, namely funding, commitment, and closing transactions, where users interact between the main Bitcoin network and the Lightning network, a second layer network that works on top of bitcoin.

Cluster 3: this cluster stands out for the lowest value of Coinbase distance among all other clusters as well as the second lowest value for the number of blocks

that it takes for outputs to be spent. The transferred amounts are also quite large. The assumption in this case is that this cluster refers to transactions that originate from mining activities, where mining rewards are high and since transactions are so close to Coinbase transactions. Most Bitcoin blocks are mined by mining pools. Once a new block is generated, the Coinbase transactions sends the respective reward along with all transaction fees to a single address for the mining pool that generated that block. The mining pool then distributes the reward and fees to its members according to their contribution in one or more new transactions in the same or in the next block. It is also a common practice among miners to move their earnings to more secure addresses soon after they receive them.

Cluster 4: this cluster is identified primarily by the lowest CoinJoin distance as well as the lowest number of blocks that outputs have remained unspent among all clusters. The total amount transacted is also relatively high and so are the fees. These quite possibly match the behavior of those who use Bitcoin for what can be described as fraudulent activity. It is common to try to disorient the origin of bitcoin transactions with CoinJoin transactions. In these, in most cases, the amounts are mixed and moved between different wallets multiple times in short periods of time. It also stands to reason that those who use Bitcoin for illicit purposes may have no problem with setting higher fees for their transactions to have them included in the next block with greater certainty.

Cluster 5: this cluster is characterized by the highest fees among all clusters as well as the highest OP_RETURN output count. It is hard to describe this cluster, but it seems to include transactions that store arbitrary data on the blockchain. Particularly for time critical applications, it is common for users to raise the fees so that the transactions are guaranteed to be included in the next block. This could be the case for services such as notarization or identity verification. However, it may also be the case that these transactions were clustered together because they happened to be placed at a time when there was a high volume of candidate transactions to be included in the next blocks in the mempool. When this happens and it is impossible to include all transactions in the mempool inside the next block, it is common practice for miners to pick the transactions that will offer them the most to gain. Hence, transactions that specify lower fees are left out in favor of ones that specify higher fees. This can quickly become a race with transaction fees ever increasing, with users trying to lure miners into including their own transactions, for as long as the mempool remains full.

Cluster 6: as previously stated, this is not actually a cluster, but rather a small collection of scattered transactions that have been removed from the k-means clustering algorithm as outliers. There is nothing that stands out with respect to these transactions. An assumption may be that these are common user transactions that for some reason sometimes transfer larger amounts with slightly larger fees than usual.

It is important to note that the majority of transactions fall into what appears to be common user transactions. Only a small proportion of transactions can be potentially identified as illicit or fraudulent. This is in contrast with other works that speculate that the volume of illicit activity is high, that is, that almost one out of four Bitcoin users was malicious and almost half of Bitcoin activity is illegal (Foley et al. 2019; Lee et al. 2020). These reports were made for transactions taking place in 2017 and Bitcoin usage patterns may have shifted. It has been reported that the number of gambling transactions in Bitcoin has dropped considerably during the last few years (Conlon and McGee 2020). Anecdotally, it has been known in the cryptocurrency communities that gambling has shifted to other alternative cryptocurrencies. In addition, the use of alternative cryptocurrencies with enhanced privacy and anonymity for transactions, such as Monero, has been increasing. Monero has become one of the leading cryptocurrencies in the market (Mensi et al. 2021) and it is estimated that approximately 25% of transactions in Monero concern illicit purposes (Miller et al. 2017). Therefore, it could be the case that illegal activities have moved to Monero or other cryptocurrencies with similar attributes. Still, a valid concern would be that it is unlikely that usage of Bitcoin has changed that dramatically and that our assumptions with respect to cluster composition have failed.

Evaluation

To test the validity of our results, we used data acquired from the walletexplorer.com database (Janda 2022). This includes addresses that have been identified as belonging to different entities. These entities are classified into five categories:

- *Exchanges* which includes known addresses from 87 online cryptocurrency exchanges.
- *Pools* which includes known addresses from 12 mining pools.
- *Services/others* which includes known addresses from 48 services related to Bitcoin, such as online wallets, coin mixers, dark web markets, and lending services, but also cryptocurrency exchanges.
- *Gambling* which includes known addresses for 41 gambling websites.
- *Old/historic* which includes known addresses by entities that do not exist anymore and that belong to all of the four previous categories.

Some of the entities in the first four categories no longer exist, but the walletexplorer.com website has not been updated in years, so they have not been moved to the "Old/historic" category. The list of addresses for existing entities is updated regularly.

A simple heuristic was used to determine that multiple addresses belong to the same entity: "Addresses are merged together if they are co-spent in one transaction. Hence, if addresses A and B are co-spent in transaction T1, and addresses B and C are co-spent in transaction T2, all addresses A, B, and C will be part of one wallet." The website's operator has interacted with each of these entities and having known at least one address that belongs to an entity, they are also able to deduce other addresses that belong to it. For each address that has been identified, the transactions that are used to make the identification are also listed.

In that dataset, we identified the transactions that occurred within the dataframe of our own dataset. These included 4246 unique transactions in the "Exchanges" category, 2508 in the "Services/others" category, 207 in the "Gambling" category, 216 in the "Pools" category, and 266 in the "Old/historic" category. Additionally, it should be noted these transaction lists are not exclusive. Several transactions are associated with addresses from multiple categories. For example, there are 121 transactions that are associated with addresses in both the "Exchanges" and "Services/others" categories, 25 in both the "Exchanges" and "Gambling" categories, while there are also seven transactions that are included in all three categories, "Exchanges," "Gambling," and "Services/others."

Out of the 4246 unique transactions included in the "Exchanges" category, our clustering analysis placed 3975 (93.6%) in cluster 2, which we also associated with online exchanges, 151 (3.6%) were placed in cluster 1 (user transactions), while 120 (2.8%) were placed in cluster 3 (mining related) (Fig. 11a).

Of the 2508 transactions in the "Services/others" category, the majority (1953, 77.9%) were placed in cluster 2, which we also associated with online services, 387 (15.4%) in cluster 4 (potentially fraudulent), 159 (6.4%) in cluster 5, which we identified with services as well, while nine (0.4%) were placed in cluster 1 (Fig. 11b).

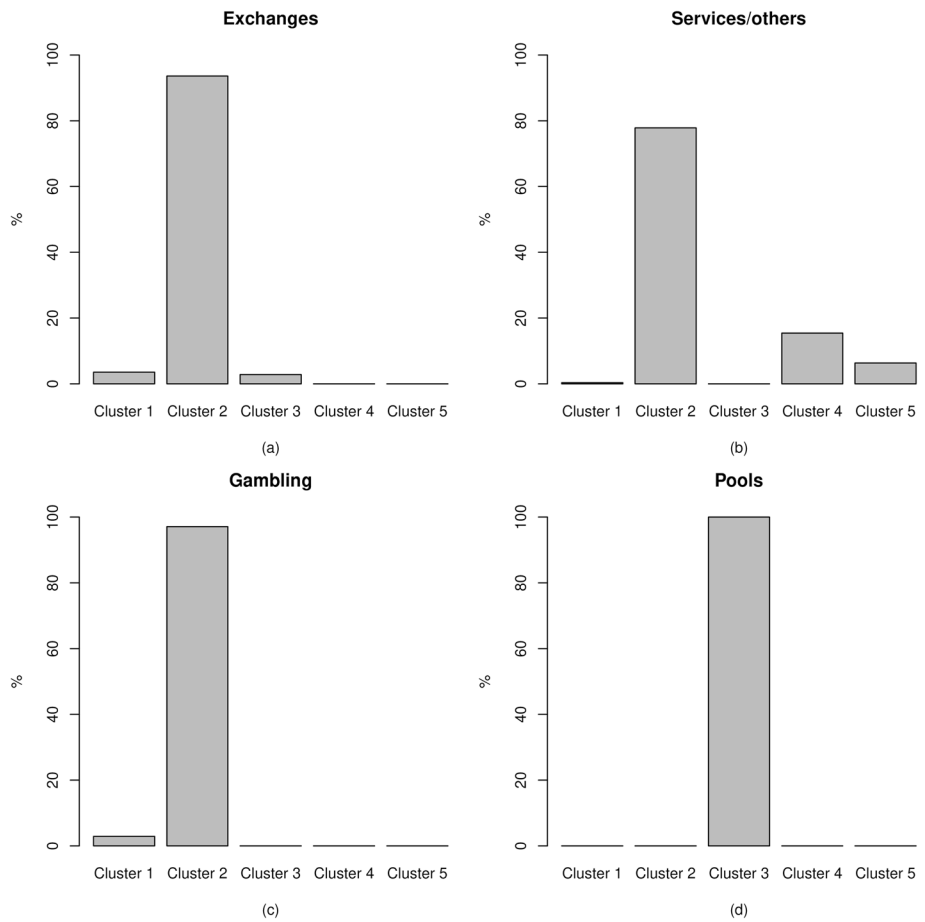

Fig. 11 Percentage of transactions in each cluster for the four entity categories found in the walletexplorer.com database. **a** Exchanges, **b** Services/others, **c** Gambling, **d** Pools

The 207 transactions in the "Gambling" category were placed almost exclusively (201, 97.1%) in cluster 2, while the rest (6, 2.9%) were placed in cluster 1 (Fig. 11c).

All 266 transactions in the "Pools" category were placed in cluster 3 (Fig. 11d).

As limited and imperfect as these results are, they indicate that our models generally agree with the data reported by walletexplorer.com. Of course, the data from walletexplorer.com should not be perceived as the ground truth as they are all based on heuristics and as has been noted, there is some overlap between categories. At the same time, there is no direct mapping from our clusters to the categories specified by walletexplorer.com. The "Gambling" category seems to have been absorbed into cluster 2 in our model, which probably also includes multiple other types of online services. However, this is a known weakness of k-means clustering; clusters with very few cases tend to get absorbed by other, larger clusters. However, this may also support our assumption that gambling related transactions have indeed mostly moved away from Bitcoin.

In summary, evaluation results reveal that our assumptions seem reasonable and our results depict an accurate overview of transactions in Bitcoin.

Conclusions

The Bitcoin blockchain provides an abundance of transaction data that carry no real-world identification properties. However, there is interest in classifying transactions with respect to their usage. Owing to the large volume of transaction data generated on a daily basis, machine learning algorithms for classification provide an indispensable tool toward this end. The lack of labeled data and the inability to produce significant amounts renders supervised algorithms unfit for such purpose.

We proposed a feature set that requires no external information other than information stored within transactions, including three novel features that have not been used in similar previous works. We implemented a data processing pipeline, starting from raw blockchain data, extracting and transforming them, until we finally analyzed them using a trimmed k-means unsupervised clustering algorithm.

Transactions that took place over the course of almost one year have been taken into account. By examining the results, transactions seem to be well-matched to the clusters they have been assigned to, albeit with some room for improvement. An attempt was made to characterize the resulting clusters, making educated guesses based on domain knowledge, according to the cluster properties. An important outcome seems to be that most transactions fall under a common usage pattern and match the behavior of ordinary users moving small amounts of bitcoin to other addresses. The number of transactions that have been identified as potentially illicit or fraudulent is relatively low.

We evaluated our results against an online database of Bitcoin addresses and their associated transactions that have been identified to be controlled by specific entities, such as online exchanges, mining pools, and lending and finance services, among others. The evaluation results indicate that there is generally a good match between the clusters these transactions were included in in our model and the categories they were assigned to in the online database. Of course, owing to the absence of more extensive ground-truth data, our assumptions about what each cluster represents may be mistaken, but we support these as reasonable and that cluster identification is probably correct, which seems to be supported by our limited evaluation results. One possible

way to further validate our results would be for the authors to engage in activity on the blockchain, such as making transactions between their accounts, exchanging fiat money on exchanges with bitcoin and moving them to their own addresses, purchasing goods by directly paying with bitcoin, trying to participate in mining operations by joining a mining pool, participating in gambling, creating applications that store data on the blockchain and using them, or even by engaging in illicit activities, or by purposefully getting infected by ransomware and paying the ransom. However, engaging in such activities is very expensive, probably prohibitively so; therefore, it would not be as easy as when Bitcoin was still very young. Even then, the amount of data gathered would be very limited compared to all the transactions that are recorded in the blockchain. It would, however, provide some additional basis for validation.

Others can use our methodology and data processing pipeline can be to conduct similar clustering analyses on different transactions of timeframes. Doing so over different timeframes may reveal shifts in transaction patterns, which may provide insights into how Bitcoin use changes, or does not change, over time. Combined with address clustering techniques, our methods may provide a way to characterize entities by their behavior on the network, even in absence of any external information. For example, it would be possible to group addresses of an otherwise unknown entity together, extract the transactions that these addresses were involved with, and then determine which clusters these transactions fall into. The distribution of transactions in these clusters may reveal significant information about the usage patterns of that entity.

Improving the performance of the clustering process could be the next step. This could be possible by deriving new features, which may bring more information into the clustering algorithms, rendering them more accurate. Additionally, a dataset of significant size that includes transactions with varying properties and which is also proven to be labeled correctly will greatly enhance the ability of using supervised machine learning algorithms for the same purpose. At the same time, it will make the evaluation of unsupervised algorithms more reliable.

Abbreviations

HD	Hierarchical deterministic
UTXO	Unspent transaction output
RPC	Remote procedure call
PCA	Principal component analysis
SSE	Sum of squared errors

Acknowledgements
Not applicable.

Author contributions
This work was conducted in collaboration with all authors. Conceptualization: GV, KK; Methodology, experimentation and data analysis: GV; Original draft writing: GV; Review and editing: GV, KK, AV; All authors read and approved the final manuscript.

Funding
This research has been co-financed by the European Union Horizon Europe Research and Innovation Programme under Grant Agreements No. 101058174 and No 101091895. The funding body did not play any role or interfere in the design of the study, the collection, analysis and interpretation of data or in writing the manuscript.

Availability of data and materials
The datasets used and/or analysed during the current study are available from the corresponding author on reasonable request.

Declarations

Competing interests
The authors declare that they have no competing interests.

Received: 1 October 2021 Accepted: 20 June 2023
Published online: 17 January 2024

References

Alqassem I, Rahwan I, Svetinovic D (2020) The anti-social system properties: Bitcoin network data analysis. IEEE Trans Syst Man Cybern Syst 50(1):21–31
Androulaki E et al (2013) Evaluating user privacy in bitcoin. In: International conference on financial cryptography and data security. Springer, pp 34–51
Arbelaitz O, Gurrutxaga I, Muguerza J, Pérez JM, Perona I (2013) An extensive comparative study of cluster validity indices. Patt Recogn 46(1):243–256
Ballis A, Drakos K (2021) The explosion in cryptocurrencies: a black hole analogy. Financ Innov 7(1):8. https://doi.org/10.1186/s40854-020-00222-0
Bartoletti M et al (2018) Data mining for detecting bitcoin ponzi schemes. In: 2018 crypto valley conference on blockchain technology (CVCBT), pp 75–84. https://doi.org/10.1109/CVCBT.2018.00014
Bistarelli S, Mercanti I, Santini F (2019) An analysis of non-standard transactions. Front Blockchain 2:7
Bitcoin wiki (2021) Bitcoin core. Accessed 28 Feb from https://en.bitcoin.it/wiki/Bitcoin_Core
Bitcoin wiki (2021) Deterministic wallet. Accessed 28 Feb from https://en.bitcoin.it/wiki/Deterministic_wallet
Bitcoin wiki (2021a) Op_return. Accessed 28 Feb from https://en.bitcoin.it/wiki/OP_RETURN
Bitcoin wiki (2021b) Segregated witness. Accessed 28 Feb from https://en.bitcoin.it/wiki/Segregated_Witness
Blockchain charts (2021) Unique addresses used. Accessed 28 Feb from https://www.blockchain.com/charts/n-unique-addresses
Bonneau J, Miller A, Clark J, Narayanan A, Kroll J A, Felten EW (May 2015) Sok: research perspectives and challenges for bitcoin and cryptocurrencies. In: 2015 IEEE symposium on security and privacy, pp 104–121. https://doi.org/10.1109/SP.2015.14
Buterin V (2021) Ethereum whitepaper, Accessed 14 Jan 2013 from https://ethereum.org/en/whitepaper/
Caprolu M et al (2021) Analysis and patterns of unknown transactions in bitcoin. In: 2021 IEEE international conference on blockchain (Blockchain)
Chen T, Tsourakakis C (2022) Antibenford subgraphs: unsupervised anomaly detection in financial networks. In: Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining, KDD '22. Association for Computing Machinery, New York, pp 2762–2770. https://doi.org/10.1145/3534678.3539100
Conlon T, McGee RJ (2020) Betting on bitcoin: does gambling volume on the blockchain explain bitcoin price changes? Econ Lett 191:108727
Cost of a 51% attack for different cryptocurrencies (2021). Accessed 27 Dec 2021 from https://www.crypto51.app/
Cuesta-Albertos J, Gordaliza A, Matrán C (1997) Trimmed k-means: an attempt to robustify quantizers. Annal Stat 25:553–576
Danovitch JH, Keil FC (2004) Should you ask a fisherman or a biologist?: Developmental shifts in ways of clustering knowledge. Child Devel 75(3):918–931
Ding C, He X (2004) K-means clustering via principal component analysis. In: Proceedings of the twenty-first international conference on Machine learning, p 29
Douglas S (2006) K-means clustering: a half-century synthesis. Br J Math Stat Psychol 59(1):1–34
Dunteman GH (1989) Principal components analysis
Foley S, Karlsen JR, Putniņš TJ (2019) Sex, drugs, and bitcoin: how much illegal activity is financed through cryptocurrencies? Rev Financ Stud 32(5):1798–1853
Han W et al (2020) Darknet and bitcoin de-anonymization: Emerging development. In: 2020 zooming innovation in consumer technologies conference (ZINC), pp 222–226
Harlev MA et al (2018) Breaking bad: de-anonymising entity types on the bitcoin blockchain using supervised machine learning. In: Proceedings of the 51st Hawaii international conference on system sciences
Harrigan M, Fretter C (2016) The unreasonable effectiveness of address clustering. In: 2016 Intl IEEE conferences on ubiquitous intelligence computing, advanced and trusted computing, scalable computing and communications, cloud and big data computing, internet of people, and smart world congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld), pp 368–373
Herrera-Joancomartí J (2014) Research and challenges on bitcoin anonymity. In: Data privacy management, autonomous spontaneous security, and security assurance. Springer, pp 3–16
Hinton GE et al (1999) Unsupervised learning: foundations of neural computation
Hirshman J, Huang Y, Macke S (2013) Unsupervised approaches to detecting anomalous behavior in the bitcoin transaction network. In: Technical report, 3rd edn. Stanford University
Janda A (2022) Bitcoin block explorer with address grouping and wallet labeling. Accessed 30 Dec 2022. https://www.walletexplorer.com/
Jolliffe IT (2002) Springer series in statistics. Princ Comp Anal 29:1403
Jourdan M et al (2018) Characterizing entities in the bitcoin blockchain. In: 2018 IEEE international conference on data mining workshops (ICDMW), pp 55–62. https://doi.org/10.1109/ICDMW.2018.00016
Kang C et al (2020) De-anonymization of the bitcoin network using address clustering. In: Zibin Z et al (eds) Blockchain and trustworthy systems. Springer, Singapore, pp 489–501

Kou G et al (2014) Evaluation of clustering algorithms for financial risk analysis using MCDM methods. Inf Sci 275:1–12
Kou G et al (2021) Bankruptcy prediction for SMES using transactional data and two-stage multiobjective feature selection. Decis Supp Syst 140:113429
Langley P et al (1994) Selection of relevant features in machine learning. Proc AAAI Fall Symp Relev 184:245–271
Lee C et al (2020) Toward detecting illegal transactions on bitcoin using machine-learning methods. In: Zibin Z et al (eds) Blockchain and trustworthy systems. Springer, Singapore, pp 520–533
Li T, Kou G, Peng Y, Philip SY (2022) An integrated cluster detection, optimization, and interpretation approach for financial data. IEEE Trans Cybern 52(12):13848–13861. https://doi.org/10.1109/TCYB.2021.3109066
Li G, Kou G, Peng Y (2022) Heterogeneous large-scale group decision making using fuzzy cluster analysis and its application to emergency response plan selection. IEEE Trans Syst Man Cybern Syst 52(6):3391–3403. https://doi.org/10.1109/TSMC.2021.3068759
Lin Y, Wu P, Hsu C, Tu I, Liao S (2019) An evaluation of bitcoin address classification based on transaction history summarization. In: 2019 IEEE international conference on blockchain and cryptocurrency (ICBC), pp 302–310
Lischke M, Fabian B (2016) Analyzing the bitcoin network: the first four years. Future Internet 8(1):7
Liu Y, Yu FR, Li X, Ji H, Leung VCM (2020) Blockchain and machine learning for communications and networking systems. IEEE Commun Surv Tutor 22(2):1392–1431
Maesa DDF et al (2016) Uncovering the bitcoin blockchain: an analysis of the full users graph. In: 2016 IEEE international conference on data science and advanced analytics (DSAA), pp 537–546
Maesa DDF et al (2018) The graph structure of bitcoin. In: International conference on complex networks and their applications. Springer, pp 547–558
Maksutov AA, Alexeev MS, Fedorova NO, Andreev DA (2019) Detection of blockchain transactions used in blockchain mixer of coin join type. In: 2019 IEEE conference of russian young researchers in electrical and electronic engineering (ElConRus), pp 274–277. IEEE
Martins S, Yang Y (2011) Introduction to bitcoins: a Pseudo-anonymous electronic currency system. In: Proceedings of the 2011 conference of the center for advanced studies on collaborative research, pp 349–350
Maurer FK (2016) A survey on approaches to anonymity in bitcoin and other cryptocurrencies. Informatik
Maxwell G (2021) Coinjoin: Bitcoin privacy for the real world. Accessed 27 Dec 2013 from https://bitcointalk.org/index.php?topic=279249.msg2983902
Meiklejohn S et al (2013) A fistful of bitcoins: characterizing payments among men with no names. In: Proceedings of the 2013 conference on internet measurement conference, pp 127–140
Mensi W, Rehman MU, Shafiullah M, Al-Yahyaee KH, Sensoy A (2021) High frequency multiscale relationships among major cryptocurrencies: portfolio management implications. Financ Innov 7(1):75
Miller A et al (2017) An empirical analysis of linkability in the monero blockchain. CoRR. arXiv:1704.04299
Monamo P, Marivate V, Twala B (2016) Unsupervised learning for robust bitcoin fraud detection. In: 2016 information security for South Africa (ISSA), pp 129–134
Nakamoto S (2008) Bitcoin: a peer-to-peer electronic cash system
National Institute of Standards and Technology (2000) Descriptions of sha-256, sha-384, and sha-512,
Neo4j graph platform (2021) Accessed 14 Jan from https://neo4j.com/
Nerurkar P et al (2020) Supervised learning model for identifying illegal activities in bitcoin. Appl Intell 51:1–20
Nerurkar P, Patel D, Busnel Y, Ludinard R, Kumari S, Khan MK (2021) Dissecting bitcoin blockchain: empirical analysis of bitcoin network (2009–2020). J Netw Comput Appl 177:102940
Pham T, Lee S (2016) Anomaly detection in bitcoin network using unsupervised learning methods. arXiv:1611.03941
Prado-Romero MA et al (2018) Discovering bitcoin mixing using anomaly detection. In: Marcelo M, Sergio V (eds) Progress in pattern recognition, image analysis, computer vision, and applications. Springer, Cham, pp 534–541
Project source code reposity at gitlab (2021) Accessed 15 Sep from https://gitlab.com/datalab-auth/blockchain/bitcoin-to-neo4j
R Core Team (2020) R: a Language and environment for statistical computing. R Foundation for Statistical Computing, Vienna
Rousseeuw PJ (1987) Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J Comput Appl Math 20:53–65
Ruppert D (2004) The elements of statistical learning: data mining, inference, and prediction
Sayadi S et al (2019) Anomaly detection model over blockchain electronic transactions. In: 2019 15th international wireless communications and mobile computing conference (IWCMC), pp 895–900. https://doi.org/10.1109/IWCMC.2019.8766765
Shafiq O (2019) Anomaly detection in blockchain. In: Master's thesis, Tampere University
Share of segwit-spending bitcoin transactions now over 50% (2021) Accessed 28 Feb from https://cointelegraph.com/news/share-of-segwit-spending-bitcoin-transactions-now-over-50
Shayegan MJ, Sabor HR, Uddin M, Chen C-L (2022) A collective anomaly detection technique to detect crypto wallet frauds on bitcoin network. Symmetry 14(2):2073
Sicignano GJ (2021) Money laundering using cryptocurrency: the case of bitcoin! Athens J Law 7:1–11
Weber M et al (2019) Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics. arXiv preprint arXiv:1908.02591
Wu Y, Tao F, Liu L, Gu J, Panneerselvam J, Zhu R, Shahzad MN (2020) A bitcoin transaction network analytic method for future blockchain forensic investigation. IEEE Trans Netw Sci Eng
Xu Jennifer J (2016) Are blockchains immune to all malicious attacks? Financ Innov 2(1):25
Zhang Y, Wang J, Luo J (2020) Heuristic-based address clustering in bitcoin. IEEE Access 8:210582–210591
Zola F, Segurola-Gil L, Bruse JL, Galar M, Orduna-Urrutia R (2022) Network traffic analysis through node behaviour classification: a graph-based approach with temporal dissection and data-level preprocessing. Comput Secur 115:102632