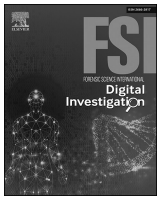
Forensic Science International: Digital Investigation 53 (2025) 301935

Contents lists available at ScienceDirect

Forensic Science International: Digital Investigation

journal homepage: www.elsevier.com/locate/fsidi



DFRWS USA 2025 - Selected Papers from the 25th Annual Digital Forensics Research Conference USA

Improved Bitcoin simulation model and address heuristic method

Yanan Gong *, Kam Pui Chow **, Siu Ming Yiu

The University of Hong Kong, Hong Kong, China

ARTICLE INFO

Keywords:
Crypto forensics
Blockchain analysis
Address clustering
Bitcoin simulation

ABSTRACT

Cryptocurrency-related crimes are on the rise and have a wide-ranging impact across various areas. To effectively combat and prevent such crimes, cryptocurrency forensics, which relies on blockchain analysis, is essential. Despite advancements in Bitcoin de-anonymization techniques, several challenges persist. The absence of authentic data labels introduces uncertainty in de-anonymization results, especially in the context of address clustering. This issue is further compounded by the development of privacy-enhancing technologies that obscure address linkages, thus undermining the reliability of outcomes as forensic evidence. To address these limitations, this study focuses on Bitcoin blockchain analysis and the improvement of address clustering. Specifically, the work presents an enhanced simulation model designed to accurately simulate real Bitcoin transactions, offering a stable platform for evaluating address clustering algorithms that utilize transaction details, thereby facilitating the assessment of the admissibility of clustering results. Meanwhile, we introduce a new heuristic algorithm aimed at identifying one-time change addresses, with experimental results demonstrating that it achieves more precise clustering outcomes than existing heuristic methods. Furthermore, our blockchain analysis reveals overarching patterns and recent changes in the Bitcoin blockchain, particularly following the introduction of the BRC-20 token.

1. Introduction

Bitcoin, as a decentralized digital currency, provides a greater degree of pseudonymity compared to traditional payment systems. However, this enhanced pseudonymity has also contributed to an increase in cryptocurrency-related cybercrimes. According to the 2024 Crypto Crime Trends report by Chainalysis, approximately $24.2 billion was funneled to illicit addresses, encompassing both stolen and illegal funds (Chainalysis Team, 2024). Cryptocurrency forensics (shortened as crypto forensics) plays a crucial role in investigating and mitigating crypto-related criminal activities. At the core of this field is the investigation and analysis of blockchain data. Specifically, Bitcoin blockchain analysis involves the systematic examination and interpretation of data recorded on the Bitcoin blockchain.

Address collection and validation are vital for blockchain forensics like fund tracing and de-anonymization. To manage the complexity of single users controlling numerous Bitcoin addresses, heuristic address clustering groups related addresses using transaction patterns. While this user behavior creates ambiguity, clustering also aids law enforcement (Reynolds and Irwin, 2017) by uncovering significant addresses and linking those belonging to the same entity, boosting investigative effectiveness. Bitcoin address clustering heuristics, which are assumption-based rules often focusing on specific transaction types, can yield inaccuracies, such as falsely grouping unrelated addresses belonging to different entities. Despite this potential for error, clustering remains an essential technique. In crypto forensics, clustering results serve primarily as guidance or supporting tools rather than standalone evidence and are typically supplemented with additional intelligence like transaction behaviors and off-chain data to improve reliability. For example, many de-anonymization studies employ this combined approach to map addresses to users for identifying real-world owners (Ermilov et al., 2017; Zhu et al., 2017; Kang et al., 2020).

Address clustering is foundational to crypto forensics, but its heuristic nature introduces risks of inaccuracies. Such errors, including misattributing addresses to incorrect entities or misdiagnosing fund flows, can compromise investigative direction, efficiency, and the overall credibility of findings. Consequently, demonstrating the validity of clustering techniques is paramount for ensuring the admissibility of resulting forensic evidence in legal proceedings. Failure to meet these standards risks the exclusion of crucial evidence, potentially hindering

* Corresponding author.
** Corresponding author.
E-mail addresses: u3556305@connect.hku.hk (Y. Gong), chow@cs.hku.hk (K.P. Chow).

https://doi.org/10.1016/j.fsidi.2025.301935

justice. While metrics like cluster size or address reduction rates are used for evaluation in some studies (Zhang et al., 2020; Chang and Svetinovic, 2018), it remains uncertain whether these indices reliably reflect true performance or merely capture side effects (like excluding relevant data), particularly given the absence of ground truth data for validation. Certain laws and regulations impose compliance constraints on the admissibility of digital evidence (Frye; Daubert; R. v. Mohan). A key factor within the Daubert criteria (Daubert) is the assessment of a method's "known or potential rate of error," offering a measure for establishing validity and reliability. This error rate serves as a measure of the limitations and potential inaccuracies of the technology, facilitating an assessment of its reliability for forensic purposes.

Due to the inherent pseudonymity of Bitcoin, definitively identifying the real user behind each address is currently infeasible, presenting a significant challenge for forensic investigators. Some public data sources that provide address labels often have limitations, such as incompleteness, lack of specific entity details, or lack of owner confirmation. This absence of ground truth data introduces uncertainty into the results of existing address clustering techniques, hindering the reliability of forensic findings. Specifically, the true owner of each address remains unknown, and a precise error rate for each heuristic address clustering method cannot currently be established, limiting the defensibility of results in court. Moreover, the increasing adoption of privacy-enhancing technologies and evolving blockchain dynamics, like new network features, have further complicated address relationship analysis, heightening existing uncertainty and presenting new challenges for crypto forensic analysis. Therefore, advancing address clustering methodologies to keep pace with these technological and behavioral shifts is crucial for achieving more accurate and dependable forensic results.

To address the challenge of unknown error rates in Bitcoin address clustering due to the lack of ground-truth data, this study introduces an enhanced simulation model tailored for Bitcoin transactions. By recording the true owner of each address, this simulator enables the verification of clustering error rates, a critical step for ensuring the reliability of forensic findings. Unlike existing models, this simulator focuses on the data layer, intentionally excluding some network-focused factors to optimize computational efficiency, given that heuristic address clustering relies primarily on transaction details. The model's enhancements are grounded in a comprehensive analysis of the structural characteristics of real-world Bitcoin blockchain transactions, ensuring its relevance to practical forensic scenarios. Furthermore, to mitigate the uncertainties caused by factors like privacy-enhancing technologies, we propose a novel heuristic algorithm for identifying one-time change addresses. This algorithm is informed by empirical analysis of real blockchain investigations, enhancing its practical applicability. Finally, this study discusses recent changes in the patterns of blockchain data following the introduction of the BRC-20 token standard.

This work offers three primary contributions:

- We develop an enhanced transaction-level simulator that accurately models Bitcoin transactions, providing a robust, controlled environment for evaluating address clustering techniques based on internal transaction details. Its capability to quantify clustering error rates facilitates assessing the reliability and limitations of these algorithms, thus supporting the admissibility of clustering results as forensic evidence.
- We propose a novel heuristic algorithm for classifying one-time change addresses, incorporating multiple qualifying criteria to better capture transaction patterns and reduce misclassifications critical in forensic analysis. Experimental results demonstrate lower error rates compared to three existing heuristics, indicating its potential to improve clustering accuracy.
- We present a comprehensive analysis of data patterns within the entire Bitcoin blockchain and compare blockchain datasets from different periods. The findings reveal changes in on-chain data characteristics and emerging trends observed in data generated following the introduction of the BRC-20 token standard. This analysis may offer reference information for the development of more effective forensic tools.

2. Related work

This section reviews relevant background work, first introducing prior research on Bitcoin analysis and then discussing existing simulation models for the Bitcoin blockchain.

2.1. Blockchain analysis

On-chain data analysis is fundamental for crypto forensics, enabling the detection of anomalies, transaction tracking, and identification of illicit activities. Early research explored foundational aspects like Bitcoin's statistical characteristics (Ron and Shamir, 2013), initial usage patterns (Badev and Chen, 2014), and methods for entity classification (Jourdan et al., 2018) and entity-level analysis (Kinkeldey et al., 2021). Additionally, the utility of on-chain data is significantly enhanced by integrating off-chain information from external sources. Examples include combining on-chain data with public off-chain records (Ermilov et al., 2017; Fleder et al., 2015) and correlating Bitcoin addresses with IP data (Koshy et al., 2014; Biryukov et al., 2014; Kang et al., 2020).

Address collection and validation are indispensable processes in blockchain analysis. Heuristic address clustering, which infers relationships between addresses based on transaction patterns, is a widely adopted approach. Commonly used heuristics include the multi-input heuristic (also known as the common spending heuristic (Ermilov et al., 2017)), utilized in numerous studies (Androulaki et al., 2013; Ron and Shamir, 2013; Meiklejohn et al., 2013), which operates on the assumption that multiple input addresses in one transaction belong to the same entity. Additionally, change address heuristics, such as the shadow heuristic (Androulaki et al., 2013) and one-time change address heuristics (Meiklejohn et al., 2013; Zhang et al., 2020; Liu et al., 2023), aim to identify change addresses among transaction outputs. However, established clustering methods are increasingly challenged as evolving blockchain transaction patterns, driven by technology and user behavior, often invalidate their underlying assumptions. This leads to diminished performance, particularly with newer, complex transactions.

2.2. Bitcoin simulation

Conducting comprehensive experiments directly on the real Bitcoin network is impractical due to its inherent complexity, time requirements, and associated costs. Consequently, researchers often employ low-cost, scalable simulation tools to explore network activities, evaluate the performance of proposed techniques, and develop novel solutions. Neudecker et al. (2015) develop a discrete-event simulation, derived from the Bitcoin application code, to assess the feasibility of network partitioning attacks. This model simplifies all cryptographic functions to achieve a balance between performance and simulation. Miller and Jansen (2015) present Shadow-Bitcoin, a framework based on the Shadow platform, which runs multi-threaded Bitcoin software within a parallel discrete-event network simulator. While this model implements a denial-of-service attack to validate its effectiveness, it has limitations that prevent it from capturing certain salient aspects of the Bitcoin network. Fadhil et al. (2016) design an event-based simulation framework to evaluate the Bitcoin Clustering Based Super Node (BCBSN) protocol. The model abstracts cryptography operations to streamline the simulation process, focusing primarily on information propagation delays within the Bitcoin network. Visualizations of Interactive, Blockchain, Extended Simulations (VIBES) (Stoykov et al., 2017) is a configurable emulator designed for massively peer-to-peer networks. This simulation tool can bypass most computationally intensive processes to enhance simulation speed. However, the model has limited

scalability and does not model specific details of blocks or transactions (Alharby and van Moorsel, 2020).

Aoki et al. (2019) demonstrate SimBlock, an event-driven simulator containing block, node, and network parameters. The simulator's construction, evaluation, and application examples primarily focus on network-level aspects, such as block propagation time and relay network participation rates. Chin et al. (2020) improve SimBlock by incorporating block mining difficulty adjustments and flexible hash rate to better model the real-world blockchain network. There is no incentive mechanism in the original SimBlock implementation. It includes a network layer and a consensus layer. Therefore, Basile et al. (2021) extend SimBlock by integrating block mining functionality, allowing it to simulate the contemporary Bitcoin blockchain network more accurately. The original SimBlock cannot recognize how Proof of Work (PoW) organizes blockchain consensus and provides actual hash computation. To address this, Mardiansyah and Sari (2022) upgrade the difficulty level based on the PoW consensus, modifying multiple elements and processes in the block mining process and visualizing the block mining process.

Alharby and van Moorsel (2020) construct BlockSim, a scalable discrete-event dynamic blockchain framework that covers network, consensus, and incentive layers. In response to the limitations of BlockSim as a local simulation on a single CPU, Agrawal et al. (2020) propose BlockSim-Net, a strengthened version of the network-based blockchain simulator. Basile et al. (2022) address limitations in BlockSim that impact blockchain performance metrics. A flexible and basic discrete-event simulator designed for multiple blockchains, also named BlockSim, is presented by Faria and Correia (2019). Fattahi et al. (2020) introduce the Merkle tree feature to BlockSim, called SIMulator for Blockchain Applications (SIMBA), aiming to reduce block verification time and improve overall simulation performance. BlockPerf, an enhanced form of BlockSim, is demonstrated by Polge et al. (2021). This updated model incorporates an incentive layer and optimizes adjustments across the existing layers. Although it excludes the contract layer, BlockPerf comprehensively covers all other layers of the blockchain system and includes the relevant metric parameters.

3. Methodology

To address previously identified research gaps, this section outlines the methodological framework for our study. Our approach consists of three stages. First, we perform an empirical analysis of the Bitcoin blockchain, examining its data to understand operational characteristics. Second, informed by these empirical findings, we describe the refinement and implementation of our simulation model. Finally, we propose a novel heuristic algorithm designed specifically to identify one-time change addresses.

3.1. Blockchain investigation

Prior knowledge of the real system is essential for the development of a robust model and its corresponding modeling methods (Murray-Smith, 2015). Before improving the simulator, this study analyzes real-world Bitcoin transactions. Existing heuristic clustering relies on filtering parameters related to transaction structure. Therefore, the blockchain investigation involves transaction and address analysis. Specifically, the transaction analysis focuses on inputs and outputs, transaction types, block transaction volume, and various fields within the transaction structure. The address analysis investigates address types and address reuse patterns. To parse the blockchain data, this study utilizes the BlockSci blockchain analysis tool (Kalodner et al., 2020) and a supplementary Bitcoin blockchain parser (Calvez). Due to the discontinuation of BlockSci support in November 2020, any blockchain data that could not be parsed by BlockSci is re-parsed using the supplementary parser. This study analyzes the first 823,786 blocks (block height 0–823,785) from the inception of Bitcoin through the end of 2023 (UTC).

3.1.1. Transaction type

The distribution of inputs and outputs across all Bitcoin transactions is first examined. As illustrated in Fig. 1-(a), transactions with a single input are the most frequent, while transactions with two outputs account for the highest proportion, followed by those with a single output. Transactions containing one to five inputs or outputs constitute over 95 % of the total, indicating that the majority of transactions fall within this range.

Based on the input and output counts, all Bitcoin transactions are categorized into four classes: consolidation, transfer, complex, and multiple payments (Cotten, 2018). Consolidation transactions aggregate Bitcoin from multiple inputs to a single output. Transfer transactions involve a single input and output. Complex transactions have multiple inputs and multiple outputs, while multiple payment transactions consist of a single input and multiple outputs. Fig. 2 displays the distribution of these transaction types. Consolidation transactions represent the smallest proportion at 5.27 %. Complex and transfer transactions are similar in proportion, constituting 19.98 % and 16.01 % of the total, respectively. Multiple payment transactions are the most prevalent, representing 58.74 % of all transactions. The transaction volume per block is also analyzed, revealing an average of 1,148 transactions per block and a clear trend of increasing transaction volumes per block over time.

3.1.2. Transaction structure

Bitcoin transactions comprise diverse fields, each with specific meanings and functions. A typical non-coinbase transaction includes inputs, outputs, a version number, and a lock time (Bitcoin.orga). This statistical analysis of transaction structure parameters excludes coinbase transactions. Each transaction is assigned a version number, indicating the corresponding authentication rule, with values typically being 1 or 2 (Bitcoin.orgb). Among the analyzed transactions, six exhibited version numbers other than 1 or 2; for example, the transaction with hash 64147d3d27268778c9d27aa434e8f270f96b2be859658950acc-de95a2f0ce79d has a version number of 0. After removing these anomalous transactions, transactions with version number 1 constitute 70.71 % of the total, with the remaining proportion attributed to transactions with version 2.

Each transaction input includes a sequence number field (Chowa; Bitcoin.orgb), which can indicate different functions. The default value, 0xFFFFFFFF, carries no special meaning. However, a sequence number of 0xFFFFFFFE implies that the lock time is enabled, allowing users to specify when a transaction should be mined. The lock time field can be set to a non-zero value representing either a specific time or a block height (Bitcoin.orgb). Transactions without lock time are more common, constituting 83.38 % of the total. A sequence number less than 0xFFFFFFFE indicates the activation of Opt-in Replace-by-Fee (RBF) (Bitcoin Core), which allows users to prioritize confirmation by increasing the transaction fee. Generally, the sequence numbers in all inputs of a transaction are identical. The probability of differing sequence numbers within a single transaction on the blockchain is less than 1 %, making it statistically insignificant. Therefore, this study considers only transactions with identical sequence numbers in all inputs. The sequence number field is categorized into three types: 0xFFFFFFFF, 0xFFFFFFFE, and values lower than 0xFFFFFFFE, with 71.90 %, 11.64 %, and 16.46 % of transactions, respectively. Segregated Witness (SegWit), related to transaction malleability (Chowb), is also analyzed. Wallets supporting SegWit can generate SegWit addresses. A flag indicating the presence of SegWit is assigned to each transaction, with the flag being true for transactions that include witness data. Notably, even coinbase transactions can incorporate SegWit. In non-coinbase transactions, the amount of SegWit-flagged transactions is 44.36 %.

3.1.3. Bitcoin address

This section investigates address type distribution and reuse patterns

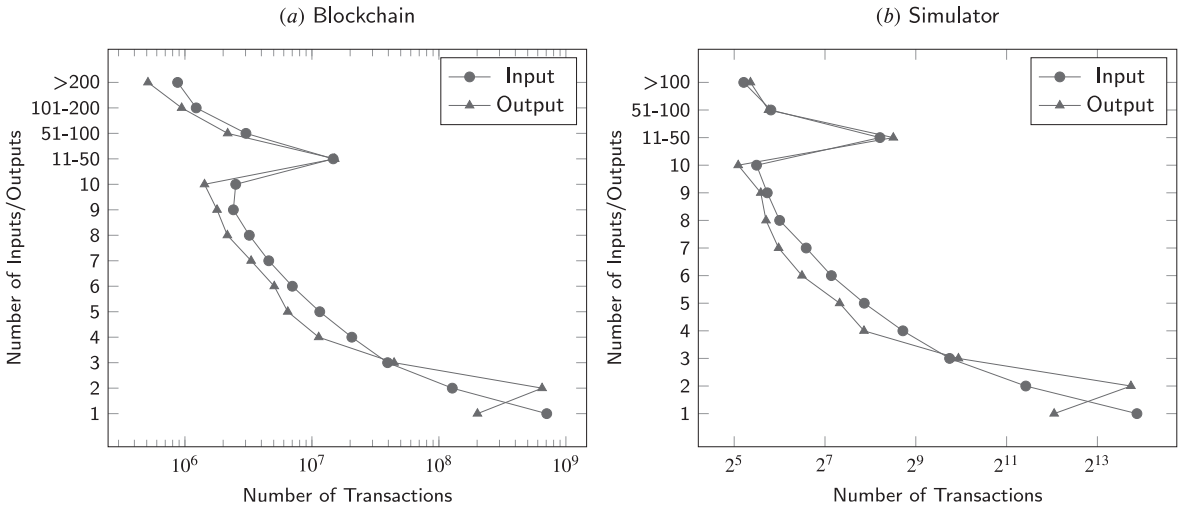


Fig. 1. Distribution of inputs and outputs.

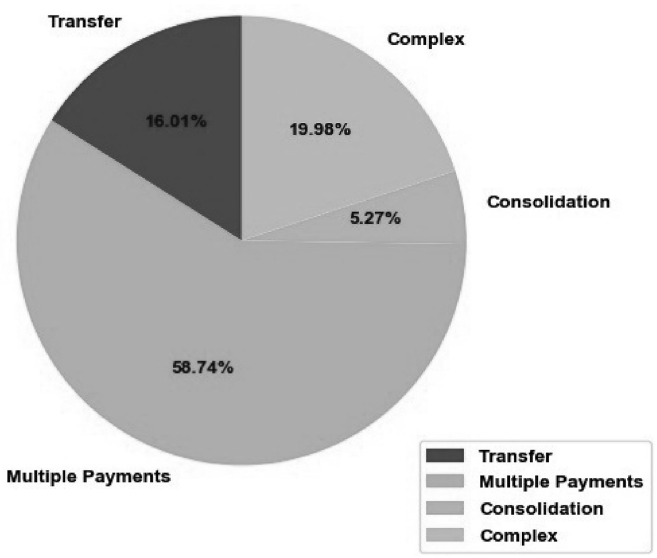


Fig. 2. Distribution of transaction types on the Bitcoin blockchain.

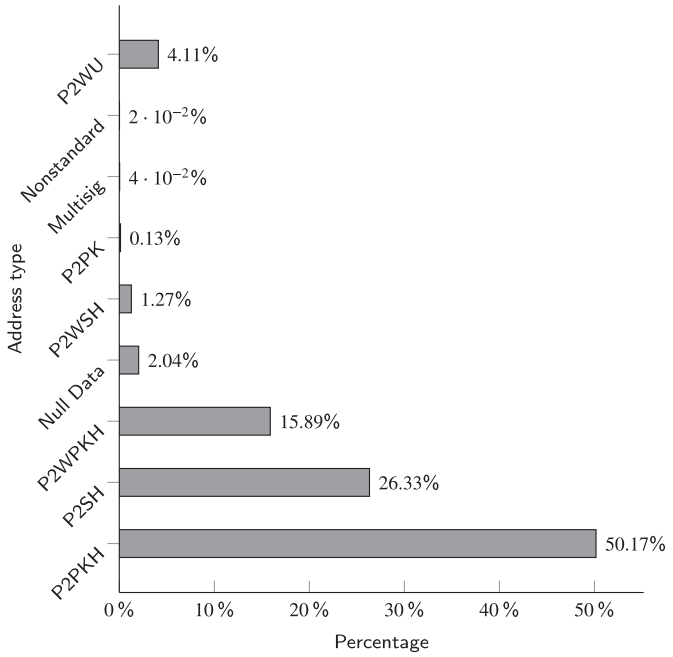


Fig. 3. Distribution of address types on the Bitcoin blockchain.

within the Bitcoin blockchain. BlockSci is employed to identify various address types. The study focuses exclusively on address types for outputs, including those from coinbase transactions. This is because transaction inputs are derived from unspent transaction outputs (UTXOs), and newly generated output addresses may subsequently serve as inputs in future transactions.

The analysis initially identifies nine address types, including Pay-to-Public-Key (P2PK), an obsolete type superseded by the more secure Pay-to-Public-Key-Hash (P2PKH) (Chowc). Null data outputs, also known as OP_RETURN, which lack a valid address string but allow for the embedding of arbitrary data on the blockchain (Chowd), are also included. Additionally, Pay-to-Witness-Unknown (P2WU) is a non-standard witness version (Bitcoin Forum). Multi-signature (Multisig) addresses constitute less than 0.05 % of the total. Non-standard addresses will be excluded from further analysis.

As illustrated in Fig. 3, the aforementioned five less prevalent address types (P2PK, Null Data, P2WU, Multisig, and Non-standard) collectively account for only 6.34 % of all output addresses, with Null Data comprising 2.04 %. User behavior is a crucial factor contributing to the low prevalence of these address types, as most users prefer the security and ease of use of the dominant address types and avoid the complexity or specialized nature of these less common types. Furthermore, the minimal proportion means its contribution to the overall dataset's characteristics is limited. Consequently, these address types were excluded from further analysis, allowing the study to focus on the dominant patterns of address usage. The study focuses on the four dominant address types: Pay-to-Witness-Public-Key-Hash (P2WPKH), P2PKH, Pay-to-Witness-Script-Hash (P2WSH), and Pay-to-Script-Hash (P2SH). Among these, P2PKH exhibits the highest prevalence, followed by P2SH, P2WPKH, and P2WSH, respectively.

To analyze address reuse, the blockchain data was segmented into annual intervals based on Coordinated Universal Time (UTC). Within each interval, a "new address" was defined as an address appearing for the first time in the blockchain, while an "old address" was defined as one appearing more than once in transaction outputs. The annual address reuse rate, presented in Fig. 4, demonstrates an upward trend during the initial four years of the study period. From 2014 onwards, the reuse rate stabilized at approximately 10 %, reaching its lowest value in 2023. Fig. 4 also presents the total number of addresses and the number

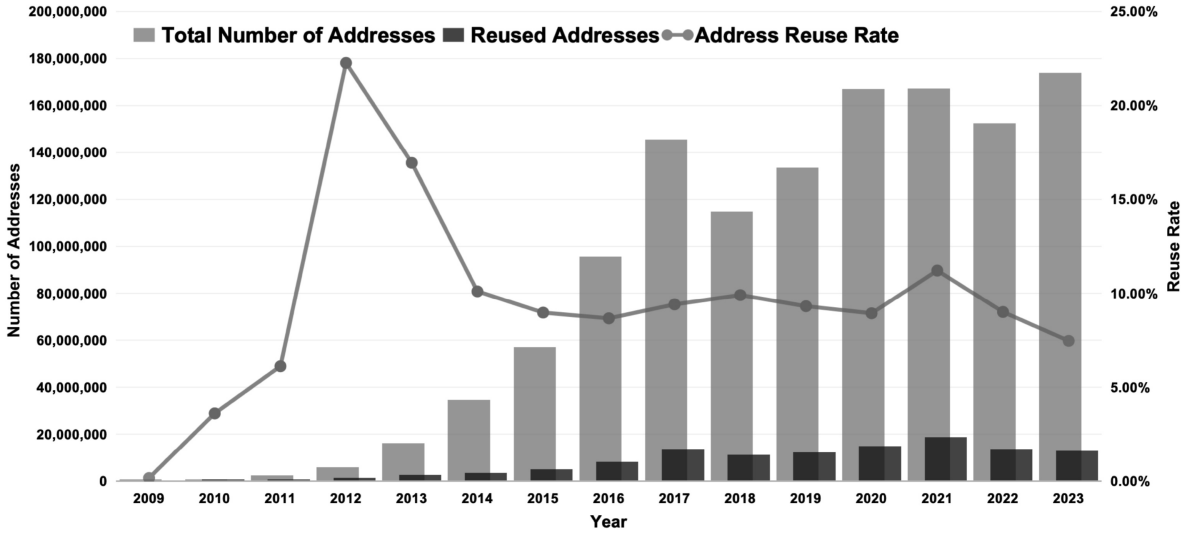


Fig. 4. Address reuse for each year on the Bitcoin blockchain.

of reused addresses for each year. In particular, while 2023 saw the generation of the highest number of new addresses, the corresponding reuse rate was comparatively low.

3.2. Simulator construction

Building upon the model proposed in (Gong et al., 2022), we have introduced modifications specifically to enable the evaluation and development of more clustering methods that leverage internal transaction details. The purpose of the aforementioned blockchain survey is to offer an in-depth understanding of the transaction structure. Therefore, the simulator's design will be adjusted based on observations from this blockchain data analysis and our specific research objectives. Within the simulator, each node operates as a full node, incorporating the essential functions of wallet management, data storage, message routing, and participation in the consensus mechanism. The underlying encryption algorithms remain unchanged from the original specification. The simulation environment is implemented using the Python programming language, and the resulting simulation data is stored upon completion for subsequent heuristic evaluation.

3.2.1. Model improvement

The transaction structure generated by the original model included a basic framework in which each transaction contained the input vector (`Vin`), input size (`vin_sz`), output vector (`Vout`), output size (`vout_sz`), used UTXOs (list of `UTXO` items), and transaction ID (`Txid`). The enhanced model includes additional functions and details in the simulated transaction structure to achieve a more realistic simulation and facilitate the evaluation of clustering techniques. These enhancements cover the fields discussed in the blockchain investigation mentioned above. As a result, the transaction structure produced by the improved simulator comprises a more comprehensive set of parameters, including transaction ID (`Txid`), version number (`version`), lock time field (`locktime`), the transaction fee (fee), input size (`vin_sz`), input vector (`Vin`), output size (`vout_sz`), output vector (`Vout`), timestamp (`timestamp`), and UTXOs used (list of `UTXO` items). Information regarding the nodes participating in the transaction and their respective sender-receiver relationships is preserved. In the input, each `Vin` contains a pointer to the relevant UTXO, a signature, a public key, a sequence number, and a witness data field. Similarly, each `Vout` in the output includes the output address, the Bitcoin amount, and the address type. Furthermore, the model incorporates adjustable settings for these newly added parameters because of the dynamic nature of the blockchain environment and changed user behavior patterns. The flexible parameter settings allow experimenters to modify these parameters in accordance with research needs and emerging blockchain patterns to create the desired simulation environment.

The original model included one-to-one and one-to-many settings. In the improved version, the one-to-one configuration is more strictly defined, ensuring that all inputs or outputs in a single transaction come from the same node. For instance, when user A sends a transaction to user B, one address in the output receives the change for user A. Although user A is the sender and user B is the receiver, each party contains only one user. However, not all output addresses necessarily belong to user B, which is the same node. This situation is classified as a one-to-many setting. Additionally, in complex transaction types involving multiple inputs and outputs, when order constraints are not applied, the one-to-many setting may also represent a many-to-one configuration. In such cases, the inputs in the transaction originate from different nodes, and all outputs belong to the same node, or all inputs belong to the same node, and the outputs originate from different nodes. These scenarios are both referred to as a one-to-many setting.

The modifications and enhancements described above are intended to increase the completeness and complexity of the simulated transactions. By achieving a more realistic simulation, the new model can be employed to evaluate address clustering techniques based on transaction structure details. The code of the simulator is available on a publicly accessible GitHub repository.[1]

3.2.2. Simulation implementation

The model allows for the configuration of the number of nodes within the simulated network and the total volume of transactions generated. During the simulation, the environment remains isolated from external interference until the simulation is completed. For internal transaction parameters, the transaction version and sequence numbers are generated according to their empirical distributions observed in the real blockchain. The lock time value is derived from the sequence number. Specifically, for transactions with the default sequence number, the lock time is disabled, and the lock time value is set to 0. When the sequence number is 0xFFFFFFFE, lock time is enabled. When the sequence number is lower than 0xFFFFFFFE, the lock time may be non-zero. The witness data field for each address is generated according to the transaction type associated with that address. For instance, Legacy addresses do not support SegWit, while P2WPKH and P2WSH addresses are classified as SegWit addresses. The transaction version and sequence

[1] Link to our GitHub Repository.

numbers are generated directly according to their empirical distributions observed in the real network.

The simulator stores all blockchain data generated during the simulation. Following the simulation, the results are saved in the following five files, which can be used for further research and verification. Instructions for each generated file can be found in the GitHub Repository.

- Simulation.log: The log records all activity information of the simulator during the entire simulation process.
- Wallet.log: It contains the wallet-related information for each node, including the confirmed UTXOs, address list, and balance. Each UTXO includes the Bitcoin amount, relevant pointer, address, address type, and index position in the original transaction.
- Transaction.log: This file stores transactions from all blocks, recording each transaction structure. The transaction format follows the JavaScript Object Notation (JSON) structure of real Bitcoin transactions. A simulated transaction is shown in Fig. 5, where nodes `(66, 85)` sent a Bitcoin transaction to nodes `(46, 87)` and `(70, 56)`. The transaction hash is `87a73ac2afc15fa4770f3b542-ce6e1ff55103d34e25cad6508714bb5ef124fdc`, containing one input and two outputs. This transaction is classified as a multi-payment transaction type. The input amount of the transaction is 8.91492563948338 BTC, with no change output. The receiving address `186PXbffPQaWB5gfDWwGiUgJzxGvutnt5t` received 2.6408770022232257 BTC. Verification of which address belongs to which recipient will require double-checking the wallet information in Wallet.log.
- Block.csv: The file saves all generated block information within the simulated network, including block hash, timestamp, transaction hash included in the block, and details related to coinbase transactions. The genesis block is located at the beginning of this file.
- Addr_type.csv: It keeps all Bitcoin addresses and corresponding address types within the simulated network. Address type labels can also be extracted by parsing the transaction structure in Transaction.log. This file is generated to facilitate any query needs.

3.3. Proposed heuristic

The dynamism of the blockchain landscape necessitates a corresponding evolution in the tools and methods used for analysis. This section introduces a novel heuristic algorithm designed to identify one-time change addresses. This algorithm utilizes a series of filtering conditions to enhance the accuracy of one-time change address identification.

Based on empirical observations and analysis of the Bitcoin blockchain, the heuristic algorithm is defined as follows for multi-output transactions with no more than six outputs:

- The transaction is not a coinbase transaction.
- The transaction does not contain a self-change address.
- All input addresses within the transaction share the same address type.
- All output addresses within the transaction are new addresses.
- Only one output address shares the same address type as the input.

The output address that satisfies all of these conditions is identified as the one-time change address for the transaction.

The rationale for these heuristic conditions is as follows. Coinbase transactions, which are used to distribute mining rewards, are excluded from this analysis. The proposed heuristic focuses on multi-output transactions with no more than six outputs, as transactions with more than five outputs constitute less than 5 % of the overall transaction volume on the Bitcoin blockchain. In a transaction, typically one output address is used to receive the sender's change. The presence of both a self-change address and a one-time change address within a single transaction is considered contradictory. Furthermore, as demonstrated by the analysis of peeling chain behavior in (Gong et al., 2023), transactions with identical input address types are more likely to originate from the same entity. The heuristic also prioritizes transactions where all output addresses are new. As shown in Fig. 4, the use of new addresses is more prevalent, and a significant number of transactions generate new addresses. Finally, if a wallet supports SegWit, it can generate SegWit addresses. Based on wallet characteristics and user habits, addresses originating from the same wallet are likely to share the same address type. Consequently, the single output address that satisfies all of the aforementioned conditions is identified as the one-time change address.

4. Experiment and analysis

This section details the experimental setup and validation procedures of the constructed simulator. Additionally, the effectiveness of the proposed heuristic algorithm is evaluated using the constructed model.

Fig. 5. The Transaction.log file generated by the simulator.

Finally, this section examines the changes in blockchain data following the introduction of the BRC-20 token standard.

4.1. Experiment setting

The simulation allows users to set the predetermined total transaction volume and node count. The total number of generated transactions will match this preset volume. Each participating node begins with a 1,000 BTC balance. To enhance realism, key parameters like input/output counts, transaction types, and block transaction volumes are based on observed network data. Address reuse is set at 10 %. Internal transaction details like version and sequence numbers follow real-world distributions, determining the lock time. Witness data is generated according to the input's address type. A random subset of nodes performs mining, with the winner receiving the reward directly into their wallet. To verify the stability and validity of the model, the experiment comprises two simulation groups:

Simulation Group 1: This group is designed to assess the stability of multiple simulation runs under consistent conditions. The node count is set to 100, and each simulation generates 15,000 transactions. The simulation is repeated six times to generate six distinct outputs.

Simulation Group 2: This group is designed to ensure that a sufficient number of transactions are generated in the simulation to accurately reflect the characteristics of the real blockchain. This group employs varying node counts and transaction volumes. First, with 100 nodes, total transaction volumes of 10,000 and 20,000 are generated, with each simulation run repeated three times (resulting in six outputs). Second, with 50 nodes, transaction volumes of 5,000, 10,000, and 15,000 are generated, with each simulation run repeated twice (resulting in six outputs). In total, 12 distinct outputs are generated from this simulation group.

4.2. Model validation

The validity of the model in simulating Bitcoin transaction behavior is verified through the validation process. This validation focuses on assessing whether the model can be considered a reasonable representation of the real-world system, given the specific research objectives (Murray-Smith, 2015). The data from two groups of experimental simulations were analyzed to generate relevant results. Due to space limitations, a random sample of simulation outputs is presented for illustrative purposes. This section presents the simulation environment with 100 nodes and 20,000 transactions. In order to evaluate the model's capacity to simulate real-world Bitcoin transactions accurately, the simulation outcomes are compared with the pattern distribution from the blockchain investigation.

4.2.1. Simulated transaction

Initially, the number of inputs and outputs in transactions is analyzed. Excluding the genesis block, the simulation generates 14 blocks, with the largest transaction volume reaching 3,802. Fig. 1-(b) illustrates the distribution ratio for inputs and outputs, revealing that 74.81 % of transactions contain one input. The trends in the simulation network are consistent with patterns observed in the real blockchain. The proportions of the four transaction types generated by the simulation are as follows: consolidation transactions account for 5.23 %, complex transactions for 15.91 %, transfer transactions for 19.96 %, and multiple payments constitute the largest proportion at 58.90 %. Regarding the distribution of address types for all outputs in the simulated network, P2PKH addresses represent more than half of the total at 58.18 %, P2SH accounts for 28.74 %, P2WPKH for 11.99 %, and P2WSH constitutes the smallest proportion at 1.09 %.

Based on observations from Simulation Group 2, an insufficient number of nodes increases the probability of transaction failure within the simulated network. Specifically, when the node count is too low, the simulation may pause during the initial phase due to the inability of a small number of nodes to generate a sufficient number of UTXOs in a timely manner. Moreover, when a sufficiently large number of nodes are configured in the simulation environment, enough transactions must be generated to serve as a representative proxy for the real blockchain. Given the substantial volume of transactions on the real blockchain, an inadequate sample size will not accurately reflect the characteristic trends observed in the real-world system.

4.2.2. Transaction details

The results of internal transaction parameters, including the transaction version number, sequence number, lock time, and SegWit flag, are presented in Table 1. A SegWit flag is assigned to each transaction based on the transaction inputs, with transactions involving SegWit addresses having the SegWit flag set to true. It is important to note that coinbase transactions are excluded from this analysis, as prior investigations of transaction structure in blockchains typically do not consider these transactions. Transactions with a version number of 1 constitute 70.63 % of the total. The portions of the three sequence number categories, default value, 0xFFFFFFFE, and values less than 0xFFFFFFFE, are 71.90 %, 11.57 %, and 16.53 %, respectively. The ratio of transactions with lock time enabled is 16.37 %. The rate of transactions with a true SegWit flag is 35.74 %. Based on these observations and comparative results, the data distribution in the simulation results closely approximates that of the real Bitcoin network, demonstrating a high level of consistency with the real-world Bitcoin system.

4.3. Heuristic algorithm evaluation

Multi-input (MI) and one-time change address (OTC) clustering are two commonly employed heuristic algorithms. In our experiments, we first establish a baseline by applying MI clustering followed by OTC clustering (represented as MI + OTC). We then evaluate the performance of our proposed new heuristic (NH) and another published heuristic (H1) (Zhang et al., 2020) by applying them subsequently to the results of this baseline MI + OTC clustering. The H1 and NH algorithms are both designed to classify all input addresses and any identified change addresses within the same transaction into a single cluster. The objective is to evaluate whether applying NH after the initial MI + OTC clustering can further refine the results, specifically by reducing the error rate and achieving more accurate identification of one-time change addresses compared to both the baseline MI + OTC results and the outcome of applying H1.

The evaluation results, obtained through six simulation runs from two experimental groups with varying combinations of node counts and transaction volumes, will be presented. These simulations include environments with 50 nodes and transaction volumes of 5,000, 10,000, and 15,000, respectively, as well as environments with 100 nodes and transaction volumes of 10,000, 15,000, and 20,000, respectively. The error rate calculation process is detailed in (Gong et al., 2022). The average error rate of MI + OTC clustering across the six datasets is 64.6805 %. This value differs from that reported in (Gong et al., 2022). A primary factor contributing to this difference is the evolution of user behavior patterns. The blockchain is a dynamic system, and user behavior is not static. The study in (Gong et al., 2022) examined the first 716,548 blocks on the Bitcoin blockchain. In that dataset, consolidation

Table 1
Distribution of internal transaction parameters in the simulated blockchain.

Parameter	Value	Percentage (%)
Transaction Version	1	70.63
Sequence Number	Default (0xFFFFFFFF)	71.90
	0xFFFFFFFE	11.57
	Less than 0xFFFFFFFE	16.53
Lock Time	Enabled	16.37
SegWit Flag	True	35.74

transactions constituted the smallest proportion, followed by transfer transactions, then complex transactions, with multi-pay transactions constituting the largest proportion. Compared with the blockchain data from the first 823,786 blocks, it is observed that multi-pay transactions, while still representing the largest proportion, have decreased in relative frequency. Conversely, the proportion of transfer transactions has increased.

The overall clustering results are visualized in Fig. 6. These results indicate that adding H1 to the MI + OTC baseline improved clustering performance by 4.8798 %. Furthermore, the MI + OTC + NH configuration (incorporating our proposed heuristic) showed a 6.1274 % improvement in performance compared to the MI + OTC + H1 result. Combining all methods (MI + OTC + H1+NH) increased overall clustering performance by a substantial 14.3249 % compared to the original MI + OTC baseline. The comparison demonstrates the synergistic effectiveness of the combined approach and highlights the contribution of our proposed heuristic (NH) in achieving more accurate clustering. To illustrate the impact from each cluster level, Fig. 7 presents detailed clustering results for a specific simulated network (100 nodes, 20,000 transactions). This figure depicts the error percentages for each true cluster compared to the clusters identified using the heuristic methods. Notably, after applying the proposed enhancements (specifically NH), the error rate for each individual cluster decreased.

Furthermore, given that these six simulations have different node counts and transaction volumes, the blockchain data generated by each simulation network varies. Therefore, slight fluctuations in the error rate are observed across different simulation results. Even in two simulations conducted under identical conditions, the generated blockchain details are not identical across different runs, resulting in fluctuations. Although the error rate fluctuates slightly, the overall value remains within a reasonable range, with no large deviations, and the outcomes demonstrate a degree of stability. The experimental findings also demonstrate that the simulator can produce relatively stable results and exhibits good stability.

4.4. Blockchain analysis

A comparison of transaction types across the two previously mentioned blockchain datasets suggests potential trends within the Bitcoin blockchain. To analyze potential changes, the blockchain data is segmented into five distinct datasets. Two datasets, denoted as `D1` and `D2`, consist of the first 716,548 blocks (block height: 0–716,547) and 772,163 blocks (block height: 0–772,162), respectively. The blockchain data utilized in prior investigations within this study is referred to as `D3`.

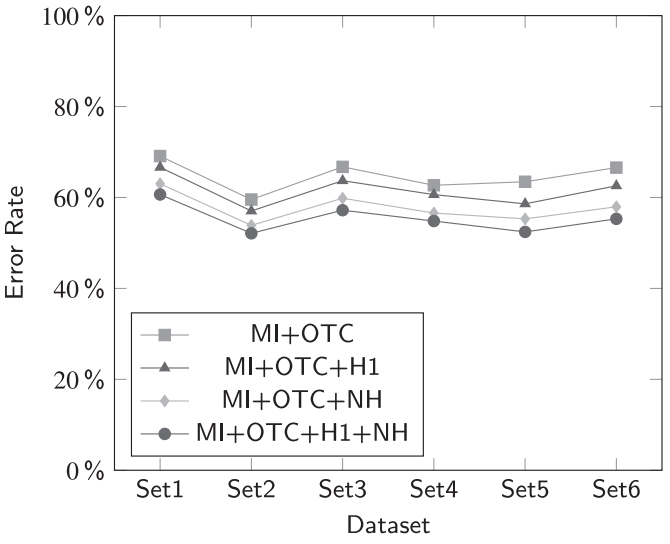


Fig. 6. The overall error rates for the address clustering.

In March 2023, the Bitcoin blockchain introduced the BRC-20 token standard (Sharma, 2022). To investigate whether the pattern of the blockchain differs after the inclusion of this new standard, we categorize the data based on the activation timeline of the BRC-20 token. Data from the beginning of the blockchain up to the BRC-20 activation (block height: 0–778,721) is labeled as `N1`, with block 778,721 representing the last block in February 2023 (UTC). Data subsequent to the BRC-20 activation (block height: 778,722–823,785) is labeled as `N2`. All relevant parameters from previous blockchain investigations are examined across all datasets. The comparative analysis reveals the following insights:

- Input and output: Fig. 8-(a) and Fig. 8-(b) illustrate the variations in input and output counts. These variations are primarily observed in transactions where the number of inputs or outputs does not exceed three. During the `N2` data period, an increase in transactions with a single input was observed, while the transaction volume with two or three inputs decreased. However, this variation had a minimal impact, as the values for `N1` and `D3` remained relatively consistent. In contrast, the differences in outputs are more pronounced. The `N2` dataset exhibits an increase in transactions with a single output, while transactions with two outputs decreased. Consequently, the proportion of transactions with one output in `D3` increased compared to the earlier datasets, `D1`, `D2`, and `N1`, while the rate of transactions with two outputs declined. When considering both inputs and outputs, the increase in transactions having a single input and a single output suggests that the prevalence of transfer transactions has risen in the new blockchain data of `N2`.
- Transaction type: The distribution results for each dataset are presented in Fig. 8-(c). Prior to the activation of BRC-20 tokens, the four transaction types exhibited similar proportion results. In the `D3` dataset, while the ranking of the four transaction categories remained unchanged, the proportion of transfer transactions increased, and the proportion of multi-pay transactions decreased. The difference between the proportions of complex and transfer transactions was 3.97 %. The `N2` dataset indicates that during the activation period, transfer transactions were the most frequent, followed by multi-pay transactions, complex transactions, and consolidation transactions, with the latter still accounting for approximately 5 %. More new transactions have only one input and one output.
- Address type: Analysis of address types using BlockSci reveals changes in their distribution. As illustrated in Fig. 8-(d), although the parsed P2WU addresses constitute a small proportion, not exceeding 5 % of the entire blockchain, their rate is gradually increasing. When parsed with the supplementary Bitcoin parser, these addresses are almost entirely identified as Pay-to-Taproot (P2TR) addresses (Wuille et al., 2020), which offer reduced fees and enhanced privacy. The increasing adoption rate of P2TR addresses is evident. While P2PKH addresses continue to dominate the blockchain, the `N2` dataset reveals a notable increase in the utilization of P2WPKH and P2TR output addresses.
- Address reuse: The address reuse rate for 2023 has dropped to the lowest level since 2014. The data for 2023 is divided into two periods based on the activation month of BRC-20. The first period, January to February, shows an address reuse rate of 9.23 %. During the second period, the address reuse frequency is 7.63 %. This data suggests that after BRC-20 was enabled, the address reuse rate in newly generated transactions decreased.
- Transaction version: To ensure accuracy in the analysis, transactions with unconventional version numbers (i.e., those not equal to 1 or 2) were excluded from each dataset. The analysis then focused on calculating the proportion of transactions with version numbers 1 and 2. The results, as depicted in Table 2, indicate a growing prevalence of transactions with version number 2 in the blockchain. This

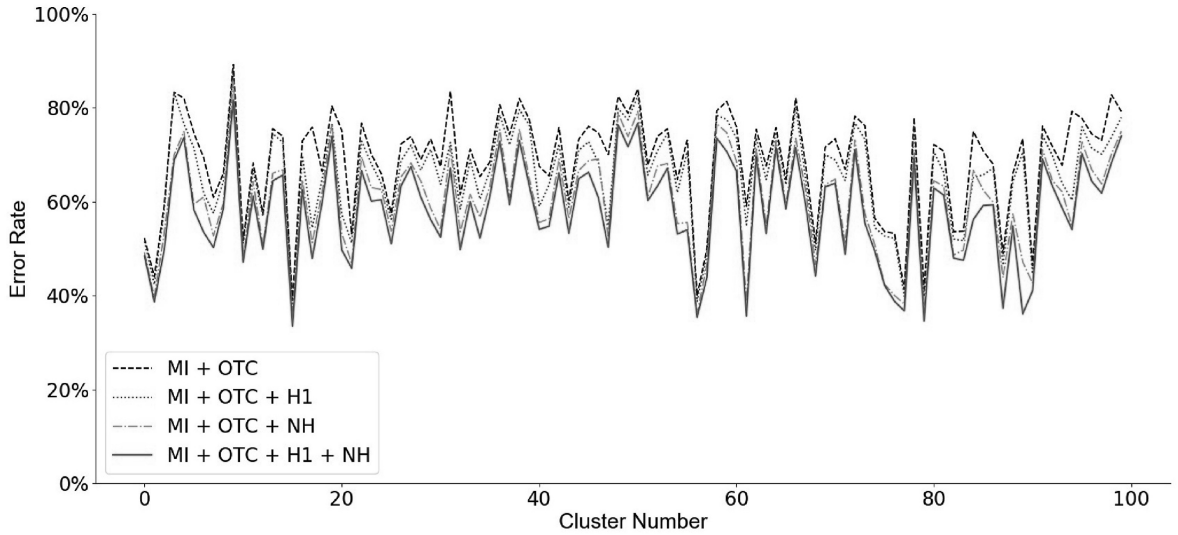


Fig. 7. Error rates of each real cluster compared to the corresponding heuristic clustering results.

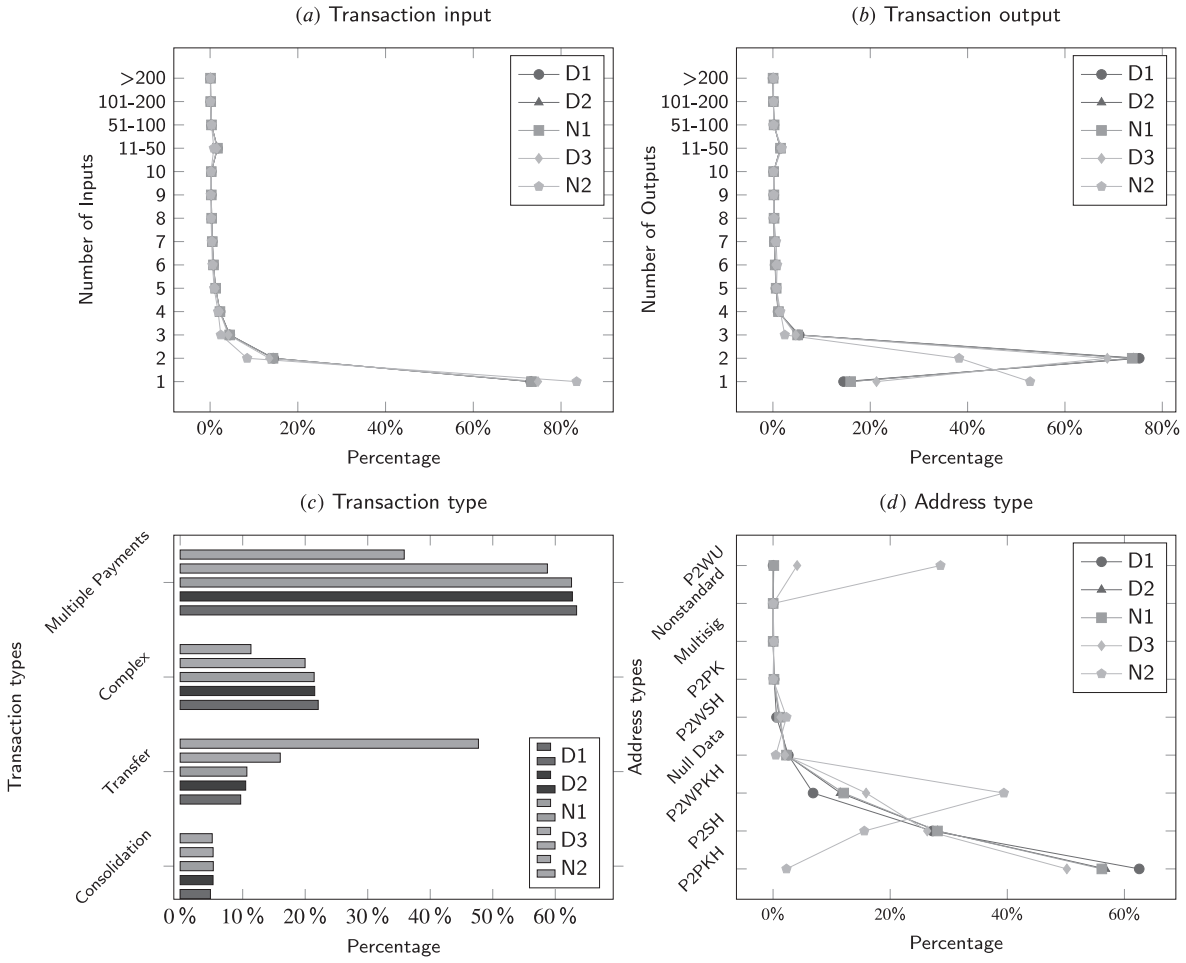


Fig. 8. Statistical distribution of five periods on the Bitcoin blockchain.

trend suggests an increasing adoption of version 2 transactions over time.

- Sequence number: The statistics in Table 2 suggest that the portion of sequence numbers in `D2` and `N1` is relatively consistent. Notably, the amount of transactions with sequence numbers less than 0xFFFFFFFE does not exceed 10 %. However, when comparing the data from `D3` and `N1`, a significant difference is observed in the distribution results, particularly in the default value category and those less than 0xFFFFFFFE. There is a marked increase in the transaction volume with sequence numbers below 0xFFFFFFFE. The results from `N2` further corroborate this trend. This increase may be attributed to the increased adoption of RBF.

Table 2
Proportion of transaction parameters (version number, SegWit usage, lock time, and sequence number) on the Bitcoin blockchain over five periods.

Dataset	Version (%)		SegWit Flag (%)		Lock Time (%)		Sequence Number (%)		
	1	2	True	False	Enabled	Disabled	0xFFFFFFFF	0xFFFFFFFE	< 0xFFFFFFFE
D1	80.21	19.79	28.64	71.36	17.75	82.25	79.75	13.68	6.57
D2	76.08	23.92	35.34	64.66	17.94	82.06	77.89	13.06	9.05
N1	75.69	24.31	36.17	63.83	17.91	82.09	77.60	12.97	9.43
D3	70.71	29.29	44.36	55.64	16.62	83.38	71.90	11.64	16.46
N2	58.89	41.11	92.97	7.03	8.91	91.09	38.02	3.70	58.28

- Lock time: The outcomes presented in Table 2 reveal that the results for the first three datasets are relatively consistent. However, slight variations are observed between the `D3` and `N1`. Beginning in March 2023, there is a slight increase in the transaction amount with lock time disabled, which aligns with the previously mentioned rise in transactions having the sequence number 0xFFFFFFFE.
- SegWit: There has been a noticeable shift in the proportion of SegWit transactions, with a growing number of transactions adopting witness data for higher efficiency, as illustrated in Table 2.

This analysis of blockchain changes, covering various time periods, allows for pairwise comparison and clearer identification of trends in recent blockchain data. Overall, the N2 dataset (blockchain data after introducing BRC-20) reveals different changes in Bitcoin transaction patterns. A main variation in input and output volumes is an increase in the simpler transactions with fewer inputs or outputs, like a rise in transfer transactions. This period also shows a growing adoption of P2TR and P2WPKH address types, indicating a trend toward enhanced privacy and fee efficiency. Additionally, the data reflects a decline in address reuse and an increased prevalence of SegWit transactions. The emergence of more transactions with sequence numbers less than 0xFFFFFFFE suggests increased use of the RBF function, while a slight decrease in lock time-enabled transactions is also observed.

This analysis details the overall changes noted on the blockchain after BRC-20 was introduced, aiming to capture the dynamic shifts occurring. It is crucial to emphasize, however, that correlation does not necessarily imply causation, particularly within the intricate Bitcoin blockchain ecosystem. Therefore, while these observations point to evolving blockchain dynamics, attributing them definitively and solely to BRC-20 requires caution and more granular research to establish causal links. This information, in turn, can potentially contribute to the development of more effective blockchain forensic tools.

5. Limitation

Despite the contributions, this study has several limitations. Firstly, while the simulation model effectively replicates real-world transaction patterns, it may not entirely cover the variability and complexity of all potential blockchain scenarios. It is essential to acknowledge that proving a model works under all possible scenarios is unrealistic (Murray-Smith, 2015). Secondly, the heuristic clustering algorithms consist of specific conditions involving observed transaction patterns and behaviors. The Bitcoin blockchain is dynamic. The evolving behavior of Bitcoin users represents a limitation. Particularly in the context of increasingly sophisticated privacy-enhancing techniques, users may alter their transaction patterns and adopt new privacy practices. This ongoing evolution makes it challenging to develop static de-anonymization techniques that remain effective over time. Thirdly, the blockchain investigation conducted in this study focuses on on-chain data. The blockchain analysis was limited to observing the behavior patterns of publicly available data on the chain.

6. Conclusion

To address the challenge of unknown error rates arising from the lack of ground-truth data, this study introduces an enhanced simulation model, specifically designed to replicate Bitcoin blockchain transactions. This model focuses on the data layer to provide a robust environment for assessing address clustering methods based on transaction details and can be modified for specific research objectives. Furthermore, the model demonstrates stability. This model records the true owner of each address, enabling the verification of clustering error rates, a critical step for ensuring the reliability of forensic findings. To mitigate the influence of dynamic factors like privacy-enhancing technologies, we propose a novel heuristic algorithm for identifying one-time change addresses. This algorithm is informed by empirical analysis of real blockchain investigations, enhancing its practical applicability. Finally, this study discusses recent trends and changes observed in Bitcoin blockchain patterns. Future research will focus on model scalability and cross-blockchain analysis.

References

Agrawal, N., Prashanthi, R., Biçer, O., Küpçü, A., 2020. Blocksim-net: A Network Based Blockchain Simulator. arXiv preprint arXiv:2011.03241.

Alharby, M., van Moorsel, A., 2020. Blocksim: an extensible simulation tool for blockchain systems. Front. Blockchain 3, 28.

Androulaki, E., Karame, G.O., Roeschlin, M., Scherer, T., Capkun, S., 2013. Evaluating user privacy in bitcoin. In: Financial Cryptography and Data Security: 17th International Conference, FC 2013, Okinawa, Japan, April 1-5, 2013, Revised Selected Papers 17. Springer, pp. 34–51.

Aoki, Y., Otsuki, K., Kaneko, T., Banno, R., Shudo, K., 2019. Simblock: a blockchain network simulator. In: IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). IEEE, pp. 325–329.

Badev, A.I., Chen, M., 2014. Bitcoin: Technical Background and Data Analysis.

Basile, M., Nardini, G., Perazzo, P., Dini, G., 2021. On improving simblock blockchain simulator. In: *2021 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, pp. 1–6.

Basile, M., Nardini, G., Perazzo, P., Dini, G., 2022. Segwit extension and improvement of the blocksim bitcoin simulator. In: *2022 IEEE International Conference on Blockchain (Blockchain)*. IEEE, pp. 115–123.

Biryukov, A., Khovratovich, D., Pustogarov, I., 2014. Deanonymisation of clients in bitcoin p2p network. In: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, pp. 15–29.

Frye v. United States, 293 F. 1013 (D.C. Cir. 1923).

Bitcoin Core, "Opt-in RBF FAQ," Available at: https://bitcoincore.org/en/faq/optin_rbf/.

Bitcoin Forum, "Shortest possible raw transaction (below 85 bytes possible)?" Available at: https://bitcointalk.org/index.php?topic=5272446.0.

Bitcoin.org, "Transactions: Null data (developer guides)," Available at: https://developer.bitcoin.org/devguide/transactions.html#null-data.

Bitcoin.org, "Transactions," Available at: https://developer.bitcoin.org/reference/transactions.html.

A. L. Calvez, "bitcoin-blockchain-parser," Available at: https://github.com/alecalve/python-bitcoin-blockchain-parser.

Chainalysis Team, 2024. 2024 Crypto Crime Trends: Illicit Activity Down as Scamming and Stolen Funds Fall, but Ransomware and Darknet Markets See Growth. Available at: https://www.chainalysis.com/blog/2024-crypto-crime-report-introduction/.

Chang, T.-H., Svetinovic, D., 2018. Improving bitcoin ownership identification using transaction patterns analysis. IEEE Transact. Syst. Man Cybernetics: Systems 50 (1), 9–20.

Chin, Z.H., Yap, T.T.V., Tan, I.K., 2020. Simulating difficulty adjustment in blockchain with simblock. In: Proceedings of the 2nd ACM International Symposium on Blockchain and Secure Critical Infrastructure, pp. 192–197.

A. Chow, "Sequence number (transactions) - bitcoin glossary," Available at: https://btcinformation.org/en/glossary/sequence-number.

A. Chow, "Bitcoin developer guide: Transaction malleability," Available at: https://btcinformation.org/en/developer-guide#transaction-malleability.

A. Chow, "How did pay-to-pubkey hash come about? what is its history?" Available at: https://bitcoin.stackexchange.com/questions/73563/how-did-pay-to-pubkey-hash-come-about-what-is-its-history.

A. Chow, "Null data (OP_RETURN) transaction - bitcoin glossary," Available at: https://btcinformation.org/en/glossary/null-data-transaction.

T. Cotten, "An overview of bitcoin transaction types," Available at: https://blog.cotten.io/an-overview-of-bitcoin-transaction-types-f22677b8e5a9, 2018.

Daubert, v., 1993. Merrell Dow Pharmaceuticals, Inc.,, 509 U.S. 579.

Ermilov, D., Panov, M., Yanovich, Y., 2017. Automatic bitcoin address clustering. In: 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA). IEEE, pp. 461–466.

Fadhil, M., Owenson, G., Adda, M., 2016. A bitcoin model for evaluation of clustering to improve propagation delay in bitcoin network. In: 2016 IEEE Intl Conference on Computational Science and Engineering (CSE) and IEEE Intl Conference on Embedded and Ubiquitous Computing (EUC) and 15th Intl Symposium on Distributed Computing and Applications for Business Engineering (DCABES). IEEE, pp. 468–475.

Faria, C., Correia, M., 2019. Blocksim: blockchain simulator. In: *2019 IEEE International Conference on Blockchain (Blockchain)*. IEEE,, pp. 439–446.

Fattahi, S.M., Makanju, A., Fard, A.M., 2020. SIMBA: an efficient simulator for blockchain applications. In: 2020 50th Annual IEEE-IFIP International Conference on Dependable Systems and Networks-Supplemental Volume (DSN-S). IEEE, pp. 51–52.

Fleder, M., Kester, M.S., Pillai, S., 2015. Bitcoin Transaction Graph Analysis. arXiv preprint arXiv:1502.01657.

Gong, Y., Chow, K.P., Yiu, S.M., Ting, H.F., 2022. Sensitivity analysis for a bitcoin simulation model. Forensic Sci. Int.: Digit. Invest. 43, 301449.

Gong, Y., Chow, K.P., Yiu, S.M., Ting, H.F., 2023. Analyzing the peeling chain patterns on the bitcoin blockchain. Forensic Sci. Int.: Digit. Invest. 46, 301614.

Jourdan, M., Blandin, S., Wynter, L., Deshpande, P., 2018. Characterizing entities in the bitcoin blockchain. In: 2018 IEEE International Conference on Data Mining Workshops (ICDMW). IEEE, pp. 55–62.

Kalodner, H., Möser, M., Lee, K., Goldfeder, S., Plattner, M., Chator, A., Narayanan, A., 2020. BlockSci: design and applications of a blockchain analysis platform. In: 29th USENIX Security Symposium (USENIX Security 20). USENIX Association, pp. 2721–2738.

Kang, C., Lee, C., Ko, K., Woo, J., Hong, J.W.-K., 2020. De-anonymization of the bitcoin network using address clustering. In: Blockchain and Trustworthy Systems: Second International Conference, BlockSys 2020, Dali, China, August 6–7, 2020, Revised Selected Papers 2. Springer, pp. 489–501.

Kinkeldey, C., Fekete, J.-D., Blascheck, T., Isenberg, P., 2021. Bitconduite: exploratory visual analysis of entity activity on the bitcoin network. IEEE Comput. Graph. Appl. 42 (1), 84–94.

Koshy, P., Koshy, D., McDaniel, P., 2014. An analysis of anonymity in bitcoin using p2p network traffic. In: Financial Cryptography and Data Security: 18th International Conference, FC 2014, Christ Church, Barbados, March 3-7, 2014, Revised Selected Papers 18. Springer, pp. 469–485.

Liu, F., Li, Z., Jia, K., Xiang, P., Zhou, A., Qi, J., Li, Z., 2023. Bitcoin address clustering based on change address improvement. IEEE Transact. Comput. Soc. Syst.

Mardiansyah, V., Sari, R.F., 2022. Simblock simulator enhancement with difficulty level algorithm based on proof-of-work consensus for lightweight blockchain. Sensors 22 (23), 9057.

Meiklejohn, S., Pomarole, M., Jordan, G., Levchenko, K., McCoy, D., Voelker, G.M., Savage, S., 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.

Miller, A., Jansen, R., 2015. Shadow-Bitcoin: Scalable simulation via direct execution of multi-threaded applications. In: 8th Workshop on Cyber Security Experimentation and Test (CSET 15).

Mohan, R.V., 1994. CanLII 80 (SCC) [1994] 2 S.C.R. 9.

Murray-Smith, D.J., 2015. "Testing and Validation of Computer Simulation Models," Simulation Foundations, Methods and Applications, pp. 233–343.

Neudecker, T., Andelfinger, P., Hartenstein, H., 2015. A simulation model for analysis of attacks on the bitcoin peer-to-peer network. In: 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM). IEEE, pp. 1327–1332.

Polge, J., Ghatpande, S., Kubler, S., Robert, J., Le Traon, Y., 2021. Blockperf: a hybrid blockchain emulator/simulator framework. IEEE Access 9, 107858–107872.

Reynolds, P., Irwin, A.S., 2017. Tracking digital footprints: anonymity within the bitcoin system. J. Money Laund. Control 20 (2), 172–189.

Ron, D., Shamir, A., 2013. Quantitative analysis of the full bitcoin transaction graph. In: *Financial Cryptography and Data Security: 17th International Conference, FC 2013, Okinawa, Japan, April 1-5, 2013, Revised Selected Papers 17*. Springer, pp. 6–24.

Sharma, S., 2022. BRC-20 Tokens: A Primer. Available at: https://research.binance.com/static/pdf/BRC-20%20Tokens%20-%20A%20Primer.pdf.

Stoykov, L., Zhang, K., Jacobsen, H.-A., 2017. Vibes: fast blockchain simulations for large-scale peer-to-peer networks. In: Proceedings of the 18th ACM/IFIP/USENIX Middleware Conference: Posters and Demos, pp. 19–20.

Wuille, P., Nick, J., Towns, A., 2020. Taproot: Segwit version 1 spending rules. Bitcoin Improvement Proposal (BIP), 341. Available at: https://github.com/bitcoin/bips/blob/master/bip-0341.mediawiki.

Zhang, Y., Wang, J., Luo, J., 2020. Heuristic-based address clustering in bitcoin. IEEE Access 8, 210582–210591.

Zhu, J., Liu, P., He, L., 2017. Mining information on bitcoin network data. In: 2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData). IEEE, pp. 999–1003.