An Empirical Evaluation of Heuristic-Based Address Clustering in Bitcoin Transaction Forensics

1st Pirzada Shayan Ahmed Kazmi
Department of Information & Communication Engineering
The Islamia University Of Bahawalpur
Bahawalpur, Pakistan
contact@shayankazmi.com

2nd Adil Zulqarnain
Department of Information & Communication Engineering
The Islamia University of Bahawalpur
Bahawalpur, Pakistan
adilzulqarnain391@gmail.com

3rd Aoun Muhammad
Department of Information & Communication Engineering
The Islamia University Of Bahawalpur
Bahawalpur, Pakistan
aoun.muhammad@iub.edu.pk

4th Sana Tariq
Department of Information & Communication Engineering
The Islamia University Of Bahawalpur
Bahawalpur, Pakistan
sana.tariq@iub.edu.pk

***Abstract*—Bitcoin offers pseudonymity in the form of cryptographic addresses. Even then we can make use of the transparency of its public transaction ledger to infer ownership patterns by using analytical techniques. In this paper, we will empirically evaluate the effectiveness of the heuristic based address clustering used for identifying entities which use the bitcoin network. In particular, we utilize the multi-input ownership heuristic and change address heuristic to build an ownership graph from actual blockchain transaction data. To construct the graph, we will be dealing with a hybrid graph representation with two types of edges: one for inferred ownership, and the other for real transaction flows. The blue edge will be used for ownership and the red edge will be used to indicate the flow of transaction between the address nodes. Heuristics will be assessed by applying graph metrics such as cluster size distribution, component analysis, and anomalously large clusters. Using the machine learning–based anomaly detection method, the Isolation Forest algorithm, we will be able to find over-clustering behavior and suspicious structural patterns that breach the heuristic assumptions. We have shown that heuristic based techniques can be used to effectively uncover ownership patterns in many situations, but they are vulnerable to over-clustering and false associations caused by privacy enhancing techniques like Coin-Join services. It provides a systematic framework to assess the address clustering problem using data-driven methods, which is based on heuristics. It also highlights its usability and drawbacks in the context of blockchain analysis.**

***Index Terms*—Blockchain Forensics, Address Clustering, Heuristic Analysis, Ownership Graph, Transaction Graph, Isolation Forest, Anomaly Detection**

I. Introduction

Bitcoin is a decentralized digital currency. It supports peer to peer transactions without the control of a central authority [1]. Bitcoin provides pseudonymity to its users by representing them by cryptographic addresses but the transactions are all public. Since transactions are transparent, we can make use of analytical techniques to try to uncover common ownership among different addresses.

One of the most commonly used techniques used in bitcoin forensics is the Address Clustering technique [2]. In this technique addresses that might be owned by a single entity are grouped into a single cluster. To do so, heuristics like "multi-input heuristic" and "change address heuristic" are implemented.

However, it is not guaranteed that these heuristics always produce accurate results. In previous studies it has been shown that these heuristics fail when privacy enhancing measures are taken [3]. There are privacy enhancing mechanisms such as CoinJoin transactions [4], [5] and large scale service behaviors including exchange batching which violate heuristic assumptions and lead to incorrect clustering. As a result the heuristic-based approach may introduce false associations.

So, we present an empirical evaluation of heuristic-based address clustering applied to Bitcoin transaction data. We will construct an ownership graph using clustering heuristics where blue edges represent ownership associations and red edges represent transaction flows. To improve clarity and scalability, we adopt a star-based representation for ownership clusters.

Furthermore, we evaluate the effectiveness of the heuristics using structural graph metrics and will make use of a machine learning–based anomaly detection approach using the Isolation Forest algorithm. Isolation Forest algorithm has helped solve several real word anomaly detection problems, thus it remains a good choice for our purpose [6]–[10]. This enables us to identify over-clustering behavior and anomalous patterns that may indicate violations of heuristic assumptions.

II. Literature Review

The transparent nature of bitcoin blockchain attracts a lot of research in analyzing transactions and trying to deanonymize bitcoin users [11], [12]. Studies done earlier have shown that although Bitcoin provides pseudonymity [2], transaction patterns can be analyzed to infer relationships between addresses and approximate user identities [13]. These works have introduced the concepts of representing blockchain data in the form of graphs, where nodes correspond to addresses

DOI: https://doi.org/10.67535/tsp.000003.025
Published under the ICACNC 2026 Proceedings

and edges represent interactions between them. To uncover common ownership of bitcoin addresses, heuristic based address clustering has been a widely used approach [14]. The assumption behind multi input heuristic is that if a single transaction includes more than one input addresses then these addresses are likely to be owned by a single entity. On the other hand, the change address heuristic looks for newly occurring addresses in the outputs of the transactions. If an output address appears for the first time in a given block of transactions, it is assumed that it is a change address and is linked to the input address. These techniques have been used a lot to construct ownership graphs and analyze user behavior within the bitcoin network. It should be noted that some research has been done in this regard which demonstrates the limitations of the said heuristics. There are methods such as CoinJoin transactions [4] which improve user privacy. CoinJoin combines input addresses from multiple transactions into a single transaction thus the assumption behind multi input heuristic gets violated. To overcome this problem, recent studies have found out analytical techniques which use graph analysis [15] and machine learning methods [16]. Graph analytics has been used in studying structural properties of transaction networks. Meanwhile machine learning techniques have been applied for anomaly detection and identification of suspicious patterns. In particular, unsupervised learning methods have shown promise in detecting irregular structures that may indicate violations of heuristic assumptions. Despite these advancements, heuristic-based clustering is still considered to be a fundamental tool in blockchain forensics because it is simple and scalable. However, there is a need for systematic evaluation frameworks that not only apply these heuristics but also critically assess their reliability in real world scenarios. In this work, we build upon existing approaches by combining the heuristic-based clustering with graph analytics and machine learning–based anomaly detection to evaluate clustering performance and identify cases of over-clustering.

III. Methodology

In this section, we will discover how to create an ownership graph from transaction data that we've gotten from the bitcoin blockchain. We will also describe the framework used to evaluate the success rate of the heuristic based address clustering. It has four stages. Data collection, address clustering, building the graph and, anomaly detection .

A. Data Collection

We use a public Bitcoin API to obtain Bitcoin transaction information. Some of the latest transaction blocks are retrieved from the blockchain and then processed. It should be noted that retrieving blocks from bitcoin blockchain is a very time consuming process, making retrieval of large datasets time inefficient [17]. At each transaction, address at input and output is extracted. Transactions which do not have valid addresses are not included in the dataset. Change address detection is also accomplished in output addresses.

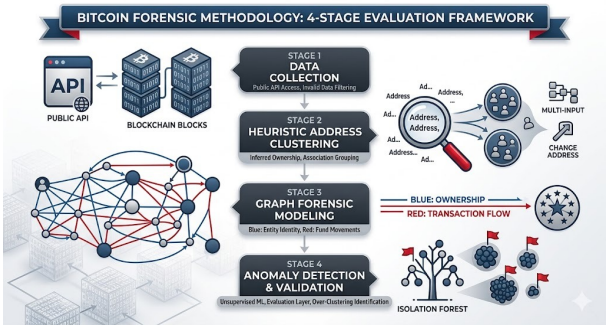

Fig. 1. Methodology Workflow

B. Heuristic Based Address Clustering

There are two mostly used heuristics to infer ownership relationships between addresses:

Multi-Input Heuristic: If a transaction has more than one address as input, then these addresses are assumed to be owned by one single entity [18].

Change-Address Heuristic: A transaction in which there are multiple outputs may have one change address. If a transaction's output has a unique address, then it is considered to be linked with the input address [19].

C. Graph Construction and Representation

We construct an ownership graph in which each node represents a bitcoin address. These nodes are connected to each other using two types of edges:

Ownership Edges (Blue):

These edges represent edges that are likely to have been created by a single entity from our heuristics, and are therefore connected to address nodes.

Transaction Flow Edges (Red):

These edges are the direction of the money flow derived from the data of transactions from the bitcoin blockchain.

D. Anomaly Detection Using Isolation Forest:

The false positive/over-clustering of addresses is known as an anomaly. To detect that we use machine learning algorithm "Isolation Forest" [20] in order to evaluate the success rate of our heuristics. We represent each cluster as a data point described by the features it has. The Isolation Forest Algorithm flags clusters which are significantly different from the majority of clusters in terms of the cluster size, ownership relation among its nodes, and behavior of its transaction flow. This will help us to detect overclustering and wrong clustering of addresses.

E. Evaluation Framework:

To assess the effectiveness of the heuristic based address clustering we use the graph metrics and the results of the anomaly detection, the evaluation criteria includes:

- Cluster size (number of addresses that the cluster has)
- Average node degree
- Internal edge density

- Ratio of transaction flow edges to ownership edges
- Frequency characteristics of output addresses

These features are employed for cluster behavior representation and further analysis.

IV. Experimental Setup

In this section, we will take a glance at the tools, data sources and implementation details used to perform the empirical evaluation of address clustering based on heuristics.

A. Data Source

A publicly available blockchain API will be used to access the Bitcoin transaction data. We receive several new transaction blocks from the blockchain. We can set the number of blocks to be pulled and this will allow us to specify scaling of the data when analysing it. For our analysis we keep the number of blocks fetched consistent.

B. Implementation Environment

The entire system is made in Python as it offers a lot of support for data processing and graph analysis. The following libraries are used:

- **Requests:**These are used to interact with the blockchain API and retrieve transaction data.
- **NetworkX:** to build and analyse the ownership graph.
- **Matplotlib:** for plotting the graph structures and cluster distributions.
- **Pandas:** Use pandas for data handling and feature extraction. Use pandas for data handling and feature extraction.
- **Scikit-learn:** to use the Isolation Forest algorithm.

C. Data Processing

This is repeated for each block being pulled, with the Input and Output addresses being pulled from each of the transactions. Data set does not contain any transactions with invalid addresses. Additionally, we count how often each output address appears in the data set to determine if there is a change in address in the data set when applying heuristics.

D. Graph Construction Parameters

The following parameters are used to construct the ownership graph:

- Only transactions where both the input and output are valid are considered .
- Ownership edges (blue) are added heuristically based on assumptions.
- For all input-output pairs, transaction flow edges (red) are added.
- To decrease the density of the clusters, star-based representation is used.

For clarity in visualization, the number of blocks to fetch is kept small during initial experiments.

E. Feature Extraction and Dataset Preparation

When the graph is constructed, we use connected component analysis on ownership edges to identify clusters. For each cluster, structural features such as size, average degree, density, and flow characteristics are computed.

These features are compiled into a dataset where each row represents a cluster and each column corresponds to a feature. This dataset serves as input to the anomaly detection model.

F. Isolation Forest Configuration

Implements the Isolation Forest algorithm with scikit-learn. The model was set with some default parameters and the contamination factor was modified to ensure the desired percentage of anomalies detected. This parameter sets the expected percentage of abnormal clusters in the data. The model is trained using the extracted cluster features in an unsupervised way, in absence of labeled data. An anomaly score is assigned to each cluster, and clusters are classified as normal or suspicious based on the value of the score.

G. Output Generation

Once the graph is built, the clusters are detected by connected component analysis on the ownership edges. The structure of each cluster is also analysed, by calculating structural properties like size, average degree, density and flow properties.

The system generates both visual and numerical outputs:

- Graph visualizations illustrating ownership and transaction flow
- Histograms representing cluster size distributions
- CSV files containing graph metrics and cluster-level features
- Anomaly detection results, including scores and labels for each cluster

All outputs are timestamped and stored locally to ensure reproducibility and facilitate further analysis.

V. Results and Evaluation

In this section, we report the results from two executions of our program, each fetching 100 latest blocks from the Bitcoin blockchain, with heuristic-based address clustering applied to each execution. The approach was evaluated using graph-based metrics, cluster-level feature analysis, and Isolation Forest-based anomaly detection.

A. Graph Construction Overview

Both executions produced ownership graphs using the multi-input and change address heuristics. The generated graphs demonstrated the structural relationships between inferred ownership associations and transaction flows.

The consistency across both executions indicates stability in the clustering heuristics and in the blockchain data collection process. The hybrid graph representation clearly distinguished inferred ownership relationships from actual transaction flows, where blue edges represented ownership associations and red edges represented transaction flows.

TABLE I
GRAPH CONSTRUCTION METRICS

Metric	Execution 1	Execution 2
Total Nodes	2,908	2,933
Ownership Edges (Blue)	2,418	2,437
Flow Edges (Red)	3,648	3,679
Total Clusters	719	726
Graph Density	0.000572	0.000567
Suspicious Clusters	72 (10.01%)	73 (10.06%)

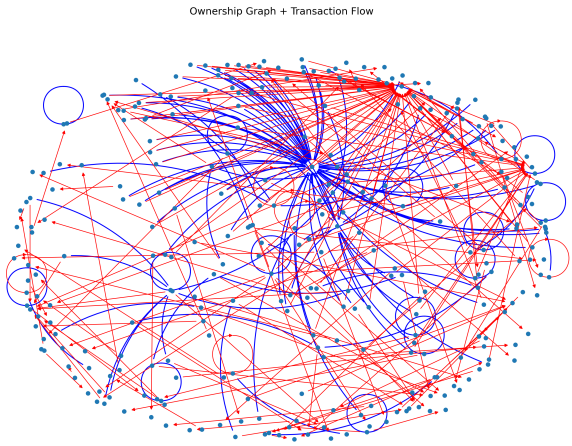

Fig. 2. Hybrid ownership graph showing ownership associations (blue) and transaction flows (red).

Visual interpretability was significantly enhanced through the use of a star-based ownership representation, which reduced graph complexity while preserving structural information.

B. Cluster Size Distribution and Structural Graph Metrics

The cluster size distributions showed that most clusters contained between 1 and 5 addresses. In the first execution, 647 out of 719 clusters (90.0%) contained five or fewer addresses. Single-address clusters accounted for 87 clusters (12.1%), while size-2 clusters formed the largest group with 396 clusters (55.1%). This distribution suggests that most entities identified by the heuristics correspond to ordinary user behavior involving a small number of addresses.

However, several significantly larger clusters were also observed. The largest cluster in the first execution contained 217 addresses, followed by clusters of sizes 103, 88, 85, 75, 73, 72, 54, 50, and 30.

The second execution exhibited a nearly identical pattern, with the largest cluster again containing 217 addresses, followed by clusters of sizes 103, 88, 85, 75, 74, 73, 54, 50, and 30. Such unusually large clusters may indicate high-activity entities such as cryptocurrency exchanges or payment processors, or they may reflect over-clustering caused by violations of heuristic assumptions.

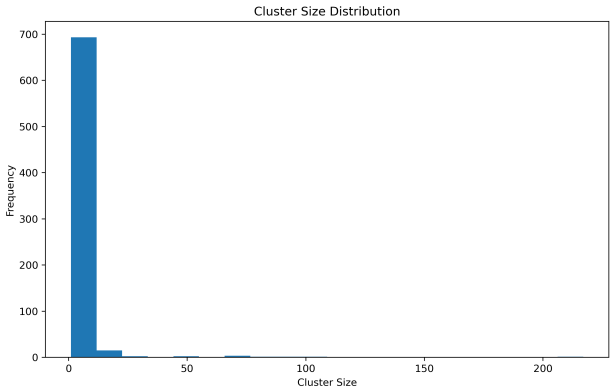

Fig. 3. Cluster size distribution for Execution 1.

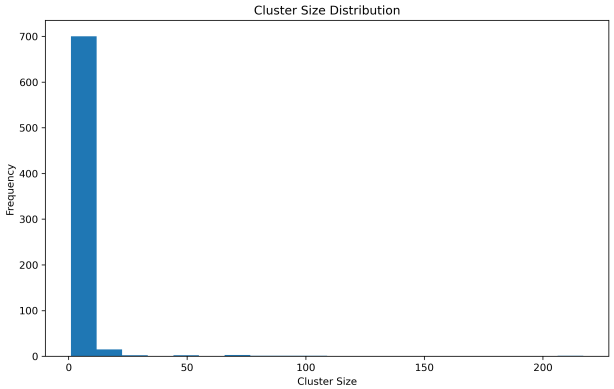

Fig. 4. Cluster size distribution for Execution 2.

Both executions were further evaluated using structural graph metrics, as shown in Table I.

The ownership graphs exhibited low overall density, which is expected in large-scale transaction networks. Nevertheless, localised regions of high connectivity were identified within certain clusters. In several size-2 clusters, density values reached 2.0, corresponding to fully connected ownership structures.

The ratio of transaction flow edges to ownership edges varied significantly across clusters, providing an informative feature for anomaly detection. Furthermore, the average node degree for ownership edges remained close to 2.0, reflecting the chain-like structure introduced by the star-based ownership representation. Suspicious clusters generally exhibited average degrees greater than 2.0, indicating more complex interconnections.

C. Detection of Anomalous Clusters

Cluster-level features including cluster size, edge count, density, average degree, and flow edge ratio were extracted and supplied to the Isolation Forest algorithm for anomaly detection.

In the first execution, 72 out of 719 clusters (10.01%) were labelled as suspicious. Similarly, the second execution identified 73 suspicious clusters out of 726 total clusters (10.06%).

The consistency of anomaly proportions across independent executions suggests stability in the anomaly detection process.

The suspicious clusters exhibited negative anomaly scores, with the most anomalous clusters obtaining scores below -0.25. The most suspicious cluster in both executions was Cluster 116, containing 217 addresses, with anomaly scores of -0.263 and -0.279 respectively. Other highly anomalous clusters included Cluster 563 (size 103, score -0.223), Cluster 117 (size 88, score -0.180), and Cluster 19 (size 85, score -0.173).

These clusters shared several characteristics:

- unusually large cluster sizes (greater than 50 addresses),
- high internal edge density relative to the global graph average,
- average node degree greater than 2.0, indicating complex inter-address connectivity.

Additionally, several small clusters were also flagged as anomalous. Examples included Cluster 135 (size 2, flow edge ratio 3.0, anomaly score -0.096), Cluster 270 (size 2, flow edge ratio 9.0, anomaly score -0.205), and Cluster 311 (size 2, flow edge ratio 9.0, anomaly score -0.218).

Although these clusters contained very few addresses, they were identified as anomalous due to extremely high flow edge ratios, which are uncommon in ordinary ownership structures. Such anomalies may indicate the presence of CoinJoin transactions or other privacy-enhancing mechanisms that generate atypical transaction patterns.

D. Evaluation of Heuristic Reliability

The results demonstrate that heuristic-based clustering can successfully group a large proportion of blockchain addresses into structurally meaningful ownership clusters. Approximately 90% of the identified clusters were classified as normal, exhibiting structural properties consistent with ordinary ownership inference.

Most normal clusters were relatively small, typically containing between 1 and 5 addresses, with density values ranging between 0.4 and 1.0 and flow edge ratios between 0.5 and 2.0.

However, the presence of approximately 10% suspicious clusters across both executions highlights inherent limitations in heuristic assumptions. These suspicious clusters potentially represent false ownership associations introduced by heuristic-based clustering. Based on the anomaly proportions, the estimated heuristic failure rate within the analysed dataset is approximately 1 in 10 clusters.

Several structural patterns corresponding to known heuristic failure scenarios were identified:

- **Large clusters (size $>$ 20):** These clusters may correspond to high-activity services such as exchanges or batching systems, but may also indicate heuristic over-clustering.
- **High flow edge ratio anomalies (flow edge ratio $>$ 3.0):** Clusters with disproportionately high transaction flow relative to ownership edges may result from incorrect change-address identification.
- **Highly dense fully connected clusters:** Clusters such as 22, 77, 149, 154, and 293 (size 2, density 2.0, anomaly score -0.007) exhibited structurally unusual connectivity patterns despite their small size.

These findings suggest that while heuristic-based clustering remains effective for many ordinary blockchain transactions, it remains vulnerable to privacy-enhancing mechanisms and atypical transaction structures that violate underlying heuristic assumptions.

VI. Conclusion

We empirically evaluated heuristic-based address clustering applied to the analysis of Bitcoin transaction data, using two independent runs both of which had 100 blocks in the blockchain. Multi-input and change address heuristics were used to build ownership graphs to approximate entity-level relationships within the Bitcoin network.

Hybrid Graph Forensic Framework

The hybrid graph representation that combined blue edges (for ownership associations) with red edges (for transaction flows) was able to separate the inferred relationships from real fund transfers. The nodes in each graph were about 2,900, the ownership edges were about 2,400, and the flow edges were about 3,600. Each execution yielded 719-726 inferred entity clusters and the overall density of the graph was approximately 0.00057. In conclusion, the star-based representation of ownership clusters greatly enhanced the interpretability while maintaining the integrity in the structure.

Clustering Effectiveness and Limitations

Structural characteristics of the majority of clusters (about 90%) were close to normal; the density values ranged from 0.4 to 1.0 and the number of addresses per cluster was between 1 and 5. About 10% of the clusters were identified as ”suspicious” by the Isolation Forest anomaly detector, however. The largest clusters (sizes 217, 103, 88, 85, and 75) were consistently seen in both executions, representing either very active entities like exchanges, or perhaps heuristic violations arising from CoinJoin transactions and/or exchange batching. In addition, though, small suspicious clusters were found with high flow edge ratios (up to 9.0) indicating that the same limitations exist with the change address heuristic when used under privacy enhancing mechanisms.

Isolation Forest as a Data-Driven Validation Layer

The Isolation Forest algorithm could be effectively used as a validation layer to detect potentially unreliable clusters while keeping the clustering results unchanged. Normal clusters had positive or near zero anomaly scores (up to 0.14), and suspicious clusters had anomaly scores between -0.007 and -0.28. The high degree of consistency between the two executions (72 suspicious clusters out of 719 total clusters, compared with 73 suspicious clusters out of 726 total clusters) reflects the repeatability (coherence) of the heuristic-based clustering and also the anomaly detection methodology.

Limitations

There are several important limitations to this study. Firstly, there is no ground truth data to directly measure the accuracy of clustering. The assessment is based on indirect criteria like structural anomalies, cluster size distributions, and consistency across executions. Second, the Isolation Forest result is dependent on the feature set used (cluster size, density, average degree, flow edge ratio) as well as the contamination parameter that may affect the percentage of anomalies marked. Thirdly, the analysis is only based on 100-block samples and may not fully reflect the long-term trend of transactions or seasonality of network activities. Fourth, the heuristics can still be manipulated by intentional privacy-preserving schemes such as CoinJoin, CoinSwap, or any other obfuscation scheme that goes against the fundamental assumptions of multi-input and change address inference.

Future Work

There are several possible lines of future research that stem from this study. First, adding more clustering heuristics (address reuse patterns, temporal transaction proximity, value-based heuristics, and others) might boost the inference accuracy and lower the false positive rate. Second, incorporating graph-based machine learning methods like Graph Neural Networks or Node2Vec embeddings could reveal more complex ownership structures than those that could be detected with heuristic rules. Thirdly, explainable AI techniques, such as SHAP value analysis, might be used on the Isolation Forest results to determine the most influential cluster features which could aid a forensic investigator in taking action based on the isolation information. Fourth, if the analysis could be extended to a larger number of blocks, encompassing thousands of blocks, and labeled datasets from known exchanges or address lists of malicious blocks could be used, then the accuracy could be directly measured. Fifth, finding and eliminating CoinJoin-like patterns prior to clustering will be a major step forward. Lastly, the framework, which would be run as an automated forensic tool with real-time alerting for anomalies, would make it more useful for blockchain investigators.

Summary

To summarize, heuristic-based address clustering is still a valuable and scalable method for blockchain forensics, but shouldn't be used as a definitive proof of ownership. The unsupervised anomaly detection is an effective tool to detect suspicious inferences, offering an intuitive way to bolster the credibility of on-chain analysis. When the same 10% suspicion rate is seen in each independent execution, it gives us a sense of the expected error rate of using these heuristics in real-world Bitcoin transaction data.

REFERENCES

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," https://bitcoin.org/bitcoin.pdf, 2008.

[2] S. Meiklejohn, M. Pomarole, G. Jordan, K. Levchenko, D. McCoy, G. M. Voelker, and S. Savage, "A fistful of bitcoins: characterizing payments among men with no names," in *Proceedings of the 2013 conference on Internet measurement conference*, 2013, pp. 127–140.

[3] Y. Gong, K.-P. Chow, H.-F. Ting, and S.-M. Yiu, "Analyzing the error rates of bitcoin clustering heuristics," in *IFIP International Conference on Digital Forensics*. Springer, 2022, pp. 187–205.

[4] F. K. Maurer, T. Neudecker, and M. Florian, "Anonymous coinjoin transactions with arbitrary values," in *2017 ieee trustcom/bigdatase/icess*. IEEE, 2017, pp. 522–529.

[5] D. Astakhin, "Keyjoin: Privacy-focused coinjoin protocol for bitcoin," Cryptology ePrint Archive, Report 2025/838, 2025. [Online]. Available: https://ia.cr/2025/838

[6] A. D. Buchdadi and A. S. M. Al-Rawahna, "Anomaly detection in open metaverse blockchain transactions using isolation forest and autoencoder neural networks," *International Journal Research on Metaverse*, vol. 2, no. 1, pp. 24–51, 2025.

[7] J. S. Tharani, E. Y. A. Charles, P. Rathore, Z. Hóu, M. Palaniswami, and V. Muthukkumarasamy, "Identifying suspicious blockchain transactions using clustering with explainability," *Distributed Ledger Technologies: Research and Practice*, 2025.

[8] A. P. Binitie, S. I. Onyemenem, N. C. Anujeonye, A. A. Ojugo, F. A. Egbokhare, and T. C. Aghaunor, "A graph-augmented isolation forest using node2vec and graphsage for mobile user behavior anomaly detection," *Journal of Computing Theories and Applications*, vol. 3, no. 3, pp. 369–383, 2026.

[9] B. E. Downey, C. K. Leung, A. G. Pazdor, R. A. Petrillo, D. Popov, and B. R. Schneider, "Anomaly detection with generalized isolation forest," in *International Conference on Advanced Information Networking and Applications*. Springer, 2024, pp. 356–368.

[10] H. Lu, "Evaluating the performance of svm, isolation forest, and dbscan for anomaly detection," in *ITM Web of Conferences*, vol. 70. EDP Sciences, 2025, p. 04012.

[11] M. Caringella, F. Violante, F. De Lucci, S. Galantucci, and M. Costantini, "Bach: A tool for analyzing blockchain transactions using address clustering heuristics," *Information*, vol. 15, no. 10, p. 589, 2024.

[12] S. Dudani, I. Baggili, D. Raymond, and R. Marchany, "The current state of cryptocurrency forensics," *Forensic Science International: Digital Investigation*, vol. 46, p. 301576, 2023.

[13] X. Yu and W. Knottenbelt, "Locard: An agentic framework for blockchain forensics," *arXiv preprint arXiv:2604.04211*, 2026.

[14] T. Ojika and S. Morishima, "A classification method for blockchain addresses considering features of address clusters," in *Proceedings of the 2024 6th Blockchain and Internet of Things Conference*, 2024, pp. 74–82.

[15] R. Cristodaro and C. J. Tessone, "Graph analytics applied to blockchain networks," in *Elgar Encyclopedia of Cryptocurrencies, Blockchain and DLT*, 2026. [Online]. Available: https://doi.org/10.4337/9781035339969.0007

[16] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 Eighth IEEE International Conference on Data Mining (ICDM)*. Pisa, Italy: IEEE, 2008, pp. 413–422.

[17] L. Zhang, T. Wang, and S. C. Liew, "Speeding up block propagation in bitcoin network: Uncoded and coded designs," *Computer Networks*, vol. 206, p. 108791, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128622000238

[18] F. Qin, Y. Wu, F. Tao, L. Liu, L. Shi, and A. J. Miller, "Multi-input address incremental clustering for the bitcoin blockchain based on petri net model analysis," *Digital Communications and Networks*, vol. 8, no. 5, pp. 680–686, 2022.

[19] F. Liu, Z. Li, K. Jia, P. Xiang, A. Zhou, J. Qi, and Z. Li, "Bitcoin address clustering based on change address improvement," *IEEE Transactions on Computational Social Systems*, vol. 11, pp. 8094–8105, 12 2024.

[20] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 Eighth IEEE International Conference on Data Mining*, 2008, pp. 413–422.