Graphaviour: Bitcoin behaviour classification based on graph topological similarities

Jon Ander Medina
Vicomtech, Basque Research and Technology Alliance (BRTA)
San Sebastian, Spain
jmedina@vicomtech.org

Mikel Gorricho
Vicomtech, Basque Research and Technology Alliance (BRTA)
San Sebastian, Spain
mgorricho@vicomtech.org

Lander Segurola
Vicomtech, Basque Research and Technology Alliance (BRTA)
San Sebastian, Spain
lsegurola@vicomtech.org

Francesco Zola
Vicomtech, Basque Research and Technology Alliance (BRTA)
San Sebastian, Spain
fzola@vicomtech.org

Raul Orduna
Vicomtech, Basque Research and Technology Alliance (BRTA)
San Sebastian, Spain
rorduna@vicomtech.org

***Abstract*—The "Graphaviour" study addresses the challenge of illicit activities in Bitcoin transactions by classifying behaviors based on graph topological similarities. Utilizing address-transaction graphs and N-step concepts, it constructs unique graphs per address to analyze Bitcoin behaviors through their structural properties, employing clustering algorithms. The methodology involves an extensive dataset of blockchain transactions, evaluated for graph-based analysis specificity. It divides the study into 1-Step and 2-Steps analyses to observe how graph depth impacts clustering accuracy versus computational load. Findings indicate that deeper graphs improve classification precision but increase computational demands, highlighting a crucial trade-off. This study not only emphasizes the importance of graph depth in analyzing Bitcoin behaviors but also suggests future research directions for diverse behavior exploration and alternative validation models. Contributing significantly to Bitcoin transaction analysis, it offers new insights into behavior classification with graph-based methodologies.**

Index Terms*—*Bitcoin, Graph Topology, Behaviour classification, Clustering, Behaviour aggregation

I. Introduction

Over the last decade, blockchain technology has dramatically reshaped the financial sector with its promise of transparency and immutability, drawing interest from a wide array of stakeholders. Yet, the same attributes that have fueled its adoption—decentralization and anonymity—have also made platforms like Bitcoin appealing to cybercriminals, creating a digital realm where illicit activities can flourish with relative impunity due to regulatory gaps.

The task of deanonymizing Bitcoin participants has thus emerged as a crucial challenge, addressed by numerous studies [10], [21], [26]. These efforts aim to enhance transparency by linking blockchain activities to real-world identities, often starting with heuristic and Open-source intelligence (OSINT) gathered from various external sources. Despite the potential of these methods, their reliance on extensive external data can make them resource-intensive. Moreover, leveraging data mining and deep learning to predict Bitcoin entity behavior and detect illegal transactions [27], as explored in the article "Cascading machine learning to attack bitcoin anonymity" where a graph construction approach based on N motifs is used, examining the number of nodes and the academic information of each generated subgraph, represents a promising but complex avenue.

Blockchain's graph-like structure, particularly the address-transaction graph, serves as the foundation for these investigations, allowing for the analysis of Bitcoin flow among addresses. Inspired by such works, our paper proposes an innovative approach that merges the address-transaction graph with N-steps analysis to craft a unique graph per Bitcoin address. This methodology enables the identification of topological similarities across behaviors, utilizing various clustering algorithms to analyze and aggregate these behaviors, with a focus on how graph depth—from 1 to 2 hops—affects clustering effectiveness and computational demand.

Preliminary results indicate that deeper graph analysis yields more accurate clustering at the cost of increased computational resources. This finding emphasizes the need for balanced feature selection to maximize clustering efficiency.

Following this introduction, the paper is structured as follows: Section II outlines the clustering algorithms and graph structures used, along with related work. Section III details our methodology, while Section IV describes the dataset, model configurations, and experiments conducted. The results and their implications are discussed in Section V, and Section VI concludes with insights and directions for future research.

II. Preliminaries

This section introduces the key concepts considered in this study, focusing on clustering within the Bitcoin transaction network and the structural dynamics of Bitcoin graphs.

Identify applicable funding agency here. If none, delete this.

ISBN:978-84-09-62140-8

A. Clustering

Our objective in clustering is to discern similar behavioral patterns within the Bitcoin transaction network's extracted subgraphs, particularly at each transaction hop. This analysis aims to identify natural clustering formations without predetermining the number of clusters, offering an authentic unsupervised exploration of the complex and diverse transaction patterns in the Bitcoin network. The selected models, DBSCAN [12], OPTICS [4], and HDBSCAN [9], are specifically chosen for their capability to adapt to the data's intrinsic structure, thereby facilitating a more genuine discovery process. Below is a detailed discussion on each model:

- **DBSCAN (Density-Based Spatial Clustering of Applications with Noise)**: DBSCAN is adept at identifying clusters as regions of high density that are separated by regions of low density. This ability is crucial for discovering clusters of arbitrary shapes within the data.
- **OPTICS (Ordering Points To Identify the Clustering Structure)**: An extension of DBSCAN, OPTICS enhances the model's capacity to detect clusters at various scales of density. This feature is particularly useful for the detailed exploration of subgraphs at different levels of granularity.
- **HDBSCAN (Hierarchical DBSCAN)**: By introducing a hierarchical approach to density-based clustering, HDBSCAN is valuable for examining subgraphs with potentially complex and layered clustering structures.

These density-based clustering models are preferred over other methods, such as agglomerative hierarchical clustering or K-means, due to their inherent flexibility in adapting to the data's natural structure and their ability to uncover clusters of diverse shapes and sizes without prior assumptions [6]. This approach ensures an unsupervised and authentic exploration of the transactional behaviors within the Bitcoin network, laying a solid foundation for a deep and precise understanding of its transactional dynamics.

B. Bitcoin Graph Structure

Transactions in Bitcoin blockchain form naturally a directed graph that can be represented by Bitcoin public key addresses and transactions (nodes) and relations (edges). In particular, edges going from an address to a transaction corresponds to incoming relations and the oposite to outgoing relations. This graph can be reconstructed directly from blockchain data by the linkage of public key addresses, as it can be seen in Figure 1. In turn, information about about relations can be retrieved too. Examples of it is the amount of money sent or timestamps, among others. This allows not only to build the directed graph, but the extraction several characteristics of it too. This graph can be modified to extract the directed address graph, where the transaction nodes are converted to different edges (subtransactions) and the remaining nodes correspond uniquely to addresses. An example is shown in Figure 2

Having uniquely defined nodes (in this case corresponding to addresses) in the graph allows to introduce the concept of n-degree node neighbourhood.

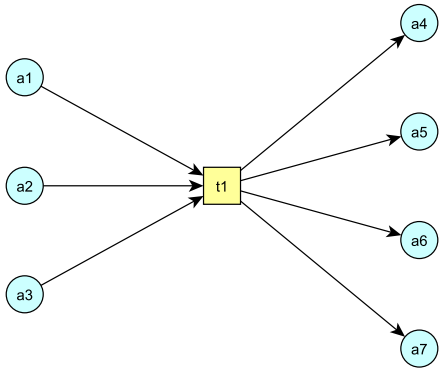

Fig. 1. Example of local bitcoin transaction graph: blue circular nodes correspond to addresses and yellow squared nodes to transaction nodes.

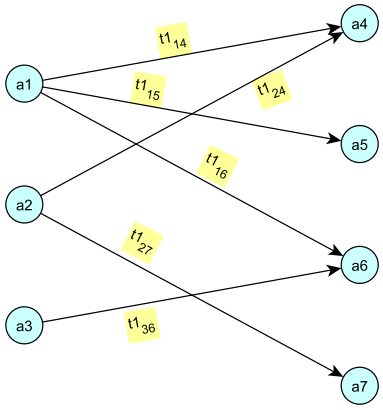

Fig. 2. Example of local bitcoin address graph: blue circular nodes correspond to addresses whereas edges correspond to subtransactions in a transaction.

Definition 1 (Path)**.** *Let V be a set of nodes, $E = \{\{u, v\} \mid u, v \in V\}$ a set of edges and $G = (V, E)$ the undirected graph built from them. Then, a path between two nodes u_0, u_n is a sequence of edges $(e_1, ..., e_n)$ where $e_1, ..., e_n \in E$, such that $e_i = \{u_{i-1}, u_i\}$ for $i = 1, .., n$. Moreover, this path is defined to have length $|\{e_1, ..., e_n\}| = n$. The set of minimal paths from u to v is defined such as the set of paths from u to v where the length of the paths is minimal.*

The definition of path allows to define n-step neighbourhood. But first, we need to provide the next lemma.

Lemma 1. *Let V be a set of nodes, $E = \{\{u, v\} \mid u, v \in V\}$ a set of edges and $G = (V, E)$ the undirected graph built from them. Let v be a node and $P_{n,v}$ the set of minimal paths of length n. For a path, $p = (e_1, ..., e_n)$ define the set $\hat{p} = \{e_1, ..., e_n\}$ and define $E' = \bigcup_{p \in P_{n,v}} \hat{p} \subseteq E$. Then, there exists $V' \subseteq V$ such that $E' = \{\{u, v\} \mid u, v \in V'\}$. Moreover*

$G_{v,n} = (V', E')$ *is a connected subgraph of* G.

Proof. The first part can be trivially proved due to the fact that $e = (u, v) \in E$, and in particular $e \in E'$, implies that $u, v \in V$. The subgraph $G_{v,n}$ connectivity is trivially proved too, due to the way it is built. □

Definition 2 (n-step neighbourhood)**.** *Let $G = (V, E)$ be an undirected graph with node set V and edge set E, and let v a node in V and $n > 0$ be a natural number. Then, the subgraph $G_{v,n} \subseteq G$ constructed in Lemma 1 is called the n-step neighbourhood of v (Figure 3 depicts an example of it. Note that $G_{n,v} \subseteq G_{n+1,v}$).*

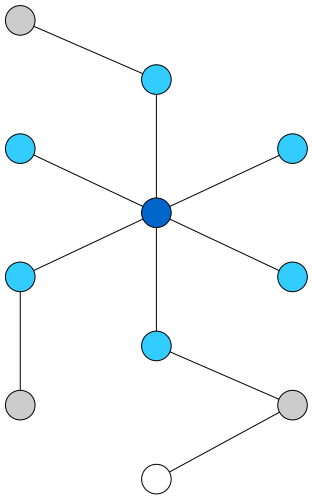

Fig. 3. 3-step neighbourhood: The darkest blue node is the central node, whereas lighter blue illustrate 1-step neighbourhood, the addition of gray nodes and corresponding edges illustrate 2-step neighbourhood and the 3-step neighbourhood is obtained by the addition of the white node.

In a directed graph, such as the address graph, directions might be omitted to construct these neighbourhoods, this is, incoming and outgoing n-steps are taken into account. An example of that can be observed in Figure 4

C. Related work

The surge in Bitcoin activity from 2020 to 2021 fueled extensive research into blockchain technology and cryptocurrency transactions, building on foundational studies. A prominent focus has been on analyzing Bitcoin transactions, transformed into graph structures, to explore user anonymity and behaviors, as seen in the work by Gaihre et al. [14] and others [5], [13]. Research has also extended to the structural analysis of these transaction graphs for predicting economic behaviors, employing feature maps, regression techniques, and neural networks, notably in Greaves and Au's 2015 study [15].

Furthermore, machine learning has been widely applied to detect patterns in transactions between wallets, with the transformation of data into Directed Acyclic Graphs (DAGs) [3], [19], [23] showcasing its effectiveness for crypto-transaction

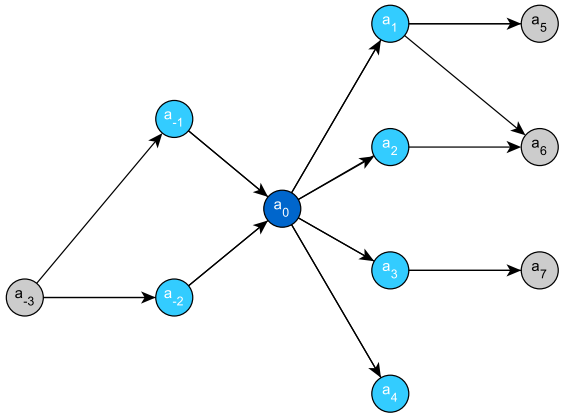

Fig. 4. 2-step neighbourhood in the address graph for address a_0.

analysis. This approach not only aids in understanding transaction behaviors but also supports the extraction and processing of features [23], highlighting the adaptability and potential of machine learning in blockchain and cryptocurrency research.

A recent approach to analyzing Bitcoin address behavior is highlighted in the study "Demystifying Bitcoin Address Behavior via Graph Neural Networks" [16] which proposes creating various graphs linking transactions (edges) to each pair of involved addresses (nodes). A tool named BAClassifier is utilized to classify Bitcoin addresses based on their transaction behaviors, as represented in these graphs, using Graph Neural Networks (GNN). However, the study does not mention augmenting graph structures using Network centrality metrics as specified. The BAClassifier achieved a precision of 96% and an F1-score of 95%, showcasing its effectiveness in classifying Bitcoin addresses based on their transaction behaviors [16]. From an economic perspective, the study by Weber et al. [24] assesses the implications of anti-money laundering regulations in the context of cryptocurrencies, with this same approach of classifying illicit transactions by applying a temporal GCN model with an F1 score of 80.6% [2].

In the case of behavior clustering approaches, there are prior cases with the use of density clustering algorithms such as OPTICS, DBSCAN, and HDBSCAN achieving a total clustering of 85.6%, 63.9%, and 68.78% respectively, based on the behavior of temporal graphs [28] for the analysis of the models' ability in noise reduction through density clusters, testing the behavior of these with $\varepsilon = [0.2, 0.5, 0.8]$ [28]. Another feature clustering approach through the use of K-means with $k = 4$, aims to represent different services within transactions, where mixing services have been identified, based on an exploratory validation of the distribution and characteristics of each cluster [22].

III. Methodology

Our methodology is designed to navigate through the process of data preparation, transformation into a graph format, extraction of subgraphs, and data analysis, specifically tailored for transactions on the Bitcoin blockchain. We begin by extracting transaction data, presented in a tabular format, between wallet pairs. This data is then transformed into a comprehensive graph structure where nodes represent individual addresses and edges symbolize transactions, incorporating both temporal and economic information to streamline the representation while preserving detailed transactional relationships.

Upon establishing this general graph, we focus on extracting subgraphs based on a predetermined central node, typically a wallet address of interest. This involves querying the graph to retrieve all related incoming and outgoing transactions as a subgraph. The extraction process is defined by an N-Step approach, where N determines the depth of connections to explore from the central node, aiming to capture a thorough transactional snapshot around it.

In the work Cascading Machine Learning to Attack Bitcoin Anonymity, where this article is presented as a new approach based on the previous method defined by Zola et al. [28], it was explained how subgraphs were generated in N-Steps based on a central address for N already declared behaviors, but using as characteristics the economic information of each of the nodes involved, as well as the number of unique nodes involved in each subgraph.

To analyze these subgraphs, we select 10 structural features to capture the transactional behaviors of the addresses, with these features detailed in Table I. Each feature contributes to a 25-element feature vector for every address analyzed, reflecting a multifaceted view of its blockchain interactions. These vectors include metrics such as degree centrality, square clustering, and eccentricity, with statistical analyses (mean, maximum, minimum, and standard deviation) applied for a comprehensive understanding. Closeness centrality is analyzed through its mean and standard deviation, while specific graph-based features are directly incorporated into the vectors as singular values.

The methodology advances by applying clustering algorithms to these feature vectors, grouping them based on similarities in the vector space. This clustering process is critical for understanding the collective behaviors of addresses. To validate the effectiveness and coherence of the clusters formed, we employ two categories of evaluation metrics: one assessing the structural quality of each cluster in relation to the entire dataset, and another focusing on the identification of outliers and the clarity of cluster definitions. This dual-evaluation approach ensures a robust analysis of the transactional dynamics within the Bitcoin blockchain.

IV. Experimental Framework

This section delineates the experimental setup utilized in our study, detailing the dataset specifications, the methodologies for data analysis, and the metrics for evaluating the performance and accuracy of our results within the Bitcoin blockchain.

Metric	Description	Value
Degree Centrality	Measures the importance of a node based on the number of links it has [18].	*v=[max, min, std, mean]*
Closeness	Measures the average closeness of a node to all other nodes in the graph [18].	*v=[std, mean]*
Transitivity	Reflects the likelihood that the adjacent vertices of a vertex are connected to each other, capturing the degree to which nodes in a graph tend to cluster together.	*v=[transativity]*
Number of Loops	Counts the number of edges that connect a node to itself [1].	*v=[number_loops]*
Number of Nodes	Counts the total nodes in the graph [1].	*v=[number_nodes]*
Number of Edges	Counts the total edges in the graph [1].	*v=[number_edges]*
Average Clustering Coefficient	The average clustering coefficient is a global measure of network segregation and reflects the clustered connections around individual nodes.	*v=[average_clustering]*
Harmonic Centrality	Summarizes the inverse of the shortest distances from a node to all other nodes in the graph [7].	*v=[max, min, std, mean]*
Square Clustering	Measures the tendency of nodes to form quadrangles in the graph [25].	*v=[max, min, std, mean]*
Barycenter	Represents the set of nodes that minimizes the sum of distances to all other nodes in the graph, essentially representing the ”center” of the graph in terms of distance.	*v=[barycenter]*
Eccentricity	Measures the maximum distance between a node and any other node in the graph.	*v=[max, min, std, mean]*
Diameter	Measures the maximum distance between any pair of nodes in the graph [17].	*v=[diameter]*

TABLE I
Explanation of Metrics

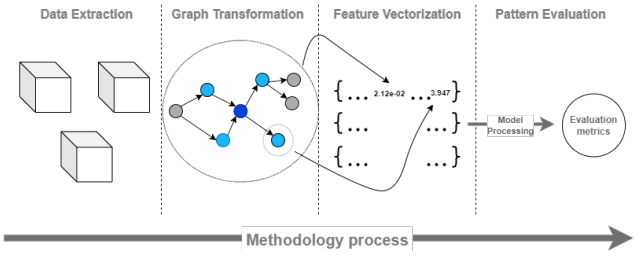

Fig. 5. Methodology Diagram

A. Dataset

In terms of the dataset, our analysis considered blockchain data up to March 2019, including a total of 566,000 blocks. Within this extensive dataset, we examined approximately 3,300,000,000 transactions and tracked over 450,000,000 unique Bitcoin addresses. To obtain the labeled data, we

leveraged WalletExplorer[1], a comprehensive platform designed for Bitcoin exploration, enabling address aggregation and wallet tagging capabilities. Subsequently, with the help of GraphSense[2], a cryptoasset analytics platform, we categorized the data into 16 distinct groups according to the Interpol[3] taxonomy. Some of the behavior were not treated due to the small amount of data related them:

- *Exchange:* Facilitates fiat-to-Bitcoin conversions and cryptocurrency trading for customers.
- *Service:* Provides Bitcoin payment solutions to various industries for seamless transaction integration.
- *Gambling:* Offers Bitcoin-based games of chance, including casinos, betting, and roulette, allowing users to wager and potentially win Bitcoin.
- *eWallet:* A digital tool for storing, managing, and conducting Bitcoin transactions securely.
- *Market:* Platforms for purchasing goods and services, including illegal ones, using Bitcoin as payment.
- *Mixing Service:* Provides anonymity to cryptocurrency transactions by mixing them with others.
- *Miner:* Collaborative miners working together to stabilize earnings while verifying transactions.
- *Loan Service:* Involves Bitcoin loans, with one party lending Bitcoin to another under agreed terms, often with interest.
- *Coinjoin:* A method of mixing tokens or coins to obscure the link between input and output in Bitcoin transactions.
- *Ransomware:* Transactions where victims pay Bitcoin to ransomware attackers in exchange for decryption keys, often in cyber extortion cases.
- *Other:* Encompasses Bitcoin transactions not fitting into predefined categories, requiring further analysis for proper classification.

Table II details the amount of data used for each behaviour for the study. In addition to the behaviours indicated, there were others that were discarded due to the small amount of data on each of them, such as ponzi_scheme, scam, BtcDice.com, Mt.Gox_Hacker or sextortion. For this study, we have excluded the Coinjoin class previously explained, as the majority of the samples are post-March 2019, and will be explored further when blockchain information is expanded, another class discarded for the study, which has been previously mentioned, is the Other class, as it is less deterministic and can group together different types of behaviors which may add noise to the study. For the study, small amounts of data for each class have been used to simplify the experiments, taking approximately 5,000 addresses for each class.

Class	*# Address*	*% Address*	*# Amount*
Exchange	12,288,433	47.84	5,000
Service	4,287,915	16.69	5,000
Gambling	3,323,767	12.94	5,000
eWallet	2,080,803	8.1	5,000
Market	2,025,747	7.89	5,000
Mixing Service	167,328	0.65	5,000
Miner	132,482	0.52	5,000
Loan service	116,900	0.46	5,000
Coinjoin	36,550	0.14	0
Ransomware	8,075	0.03	5,000
Other	1,156,955	4.5	0
Total	**25,624,955**	**100**	**45,000**

TABLE II
OVERVIEW OF DATA USED FOR THIS STUDY

B. Evaluation Metrics

Evaluating the quality of generated clusters is a challenging task, especially in the context of undefined groupings based on the structural behavior of each subgraph, as the groupings are based on this information where examples of similar labels can be joined in different clusters. For this reason, we first propose to use three common metrics, i.e., *Silhouette Coefficient, Calinski-Harabasz Index*, and *Davies-Bouldin Index* that help to evaluate the structure of clusters in the vector space.

More specifically, the *Silhouette Coefficient measures (or SC)* the cohesion and separation of clusters, providing an indication of the distance between the resulting clusters [20]. It takes values between -1 and 1, where a high value indicates that the point is well clustered. The *Calinski-Harabasz Index (or CH Index)* evaluates the dispersion within and between clusters [8]. The values can range from near 0 to very high values (no upper limit), with higher values being preferable as they indicate denser and well-separated clusters. The *Davies-Bouldin Index (or DB Index)* evaluates the average of the similarities between each cluster with its most similar cluster, where similarity is the ratio of the distance between clusters and the sum of the dispersions within the clusters [11]. The values oscillate between 0 and higher values, with lower values being preferable as they indicate a better separation between clusters.

Then, we define three new metrics that we called *Homogeneous Cluster (or HC), Noisy Cluster (or NC)* and *Outliers Cluster (or OC)*, for evaluating the cluster composition based on the grouped labels.

The objective of these metrics is to classify the quality of clusters based on their heterogeneity, with those having a higher diversity of labels in their composition being rated lower. To define these metrics, we have created a threshold that ensures the predominant class has at least 10% more samples than the second most populous class. Clusters meeting this sample volume that exceeds the threshold are grouped into *Homogeneous Clusters*. Those that do not reach this threshold are grouped into *Noisy Clusters*, which also includes clusters with less than three samples in their composition. Lastly, those grouped in cluster -1 by the employed algorithms are classified as *Outlier Clusters*.

To enhance confidence in the metrics generated to evaluate the label-based composition of each cluster, we have introduced two additional metrics termed as *"Cluster Confidence"*. These metrics assess the consistency of clusters identified as *Homogeneous Clusters*. They involve calculating the *Mean*

[1]https://www.walletexplorer.com/
[2]https://graphsense.info/
[3]https://interpol-innovation-centre.github.io/DW-VA-Taxonomy/

value of the percentage of samples from the specified class in each cluster relative to the total samples of that cluster, as well as the *Standard Deviation* of these calculated values.

C. Experiments

The main objective of the experiments is to understand the clustering behavior, performance, and effectiveness of these models in segmenting similar behaviors within the subgraphs, without requiring a predefined number of clusters. To obtain the subgraphs for each recognized entity in both 1-Step and 2-Step categories, we used a machine dedicated to the generation and preprocessing of subgraphs, as well as the analysis and execution of the models. The specifications of the machine defined 3T of disk memory, 64GB of RAM, and 40 available CPU cores. For the preprocessing and generation of the subgraphs, the capacity of these cores was used, parallelizing both processes.

The experiments are divided into 1-Step and 2-Step categories because even though the same perspectives are used, an individual approach is necessary for each new depth level within the subgraphs. The time for extraction and preprocessing of each subgraph varies depending on the load on the database, the volume of the subgraph, and its depth. In the case of 1-Step graphs, the fastest extracted subgraph took about 3 seconds, and the longest took approximately 2 hours. For the 2-Step subgraphs, the smallest took around 7 seconds but could take more than a day for the extraction and preprocessing of those with a substantial volume of nodes. The growth in preprocessing time is exponential and difficult to calculate precisely.

To conduct various experiments, we established an initial configuration for each model used in this paper. Table III provides a comprehensive explanation of the chosen parameter configurations for these models.

Model	Configuration
DBSCAN1	min_samples: 5 (ϵ: 0.5)
DBSCAN2	min_samples: 10 (ϵ: 0.5)
DBSCAN3	min_samples: 20 (ϵ: 0.5)
HDBSCAN1	min_samples: 5
HDBSCAN2	min_samples: 10
HDBSCAN3	min_samples: 20
OPTICS1	min_cluster_size: None (ϵ: 0.5)
OPTICS2	min_cluster_size: 10 (ϵ: 0.5)
OPTICS3	min_cluster_size:20 (ϵ: 0.5)

TABLE III
MODEL PARAMETERS CONFIGURATION

1) 1-Step: In the 1-Step experiments, two different perspectives were employed for the subgraphs generated. These perspectives were based on the number of structural graph features to extract, and they aimed to provide a more comprehensive understanding of the subgraph behavior.

The initial approach considered only 10 metrics which included Degree Centrality, Closeness Centrality, Transitivity, Number of Loops, Number of Nodes, and Number of Edges. Another approach was defined to provide a more detailed insight into the sub-graph structure, incorporating the rest of metrics providing 25, all detailed on Table I.

Using the feature vectors generated by the reduced metrics, 3 density-based clustering models were implemented to perform clustering while exploring different parameters as shown on Table III to determine an optimal model for the 1-Step. To validate the formation of clusters in terms of their structural composition, we used metrics such as Silhouette Coefficient (SC), Calinski-Harabasz Index (CH Index), and Davies-Bouldin Index (DB Index).

After cluster validation with structural composition metrics, we decided to validate the formation of the clusters using label-based metrics, specifically applied to data generated with extended features. This decision was made because we aimed to understand the composition of the clusters from the perspective of labels, and extended features offer more structural information and a better understanding of label-level composition.

2) 2-Steps: In the 2-Step experiments, a similar approach was followed as in the 1-Step experiments, but this time, we extended the analysis to a deeper level of subgraph features. Based on the results obtained from the experiments conducted with the reduced structural features, which are available in Table I, the decision has been made to utilize the approach with 25 characteristics with all detailed model configurations on Table III.

Each of the best configurations per model, as determined by the cluster evaluation metrics, has been further evaluated using label-based metrics detailed on Subsection IV-B.

Additionally, two metrics based on the mean value of the winning label percentages in their cluster and the standard deviation of the obtained values have been applied. The goal is to determine the internal quality of the cluster composition, providing more information about their homogeneity and confidence in the label-based metrics.

V. RESULTS

A. 1-Step Analysis

The results for 1-Step Sub-Graph experiments are shown on Table IV with previous detailed points on Subsection IV-C.

Model	10 Features			25 Features		
	SC	CH Inx	DB Inx	SC	CH Inx	DB Inx
DBSCAN1	0.9509	294.9165	1.2409	0.9463	44.5253	1.2859
DBSCAN2	0.9473	321.1931	1.6354	0.9247	54.2147	1.2593
DBSCAN3	**0.9440**	**334.4038**	**1.0986**	0.8945	48.3408	1.2534
OPTICS1	0.9707	128.9586	2.2995	**0.9568**	**110.8129**	**1.5871**
OPTICS2	0.9471	123.5622	4.6214	0.9340	56.3546	1.5305
OPTICS3	0.9260	88.8228	2.2883	0.8995	75.5074	1.3032
HDBSCAN1	0.2532	89.0173	2.1068	0.1510	75.9186	1.4392
HDBSCAN2	0.2556	52.7525	1.8184	0.1549	40.1424	1.4164
HDBSCAN3	0.2676	68.6815	1.8221	0.1426	48.8047	1.3862

TABLE IV
PERFORMANCE METRICS OF CLUSTERING ALGORITHMS ON 1-STEP SUB-GRAPHS.

As demonstrated in the presented results, the models executed with extended features showcase superior performance across all the models used, rendering the reduced approach

inadequate due to its limited information regarding the behavior of each sub-graph. In the case of extended features, the OPTICS1 model emerged as the most robust, while in the case of reduced metrics, the DBSCAN3 model demonstrated superior robustness when examining cluster formation metrics.

Model	Features	HC	NC	OC	Mean	Standard Deviation
DBSCAN3	10	28	14	736	0.2859	0.1062
OPTICS1	**25**	**257**	**52**	**823**	**0.4881**	**0.1933**

TABLE V
COMPARISON OF THE BEST ALGORITHMS FOR THE TWO APPROACHES TO 1-STEP GRAPH FEATURE CHARACTERISTICS.

To determine which of the two approaches, based on the number of features to be extracted from graphs, is superior, we evaluated the best models using label-based evaluation metrics and their validation metrics. As we can see in Table V, the model based on extended features turns out to be the one that can generate the most clusters, with greater homogeneity in the data that constitute them. It establishes that the average cluster has a predominant class that makes up at least 0.4881 of the samples comprising it, with a standard deviation of 0.1933. In contrast, the approach with 10 features offers lower performance and fewer clusters, creating very heterogeneous clusters in terms of different behaviors or labels, forming 28 versus 257 in the case of extended metrics, as seen in Table V.

B. 2-Steps Analysis

The predefined models in the configuration Table IV have been applied in this perspective, and the results of the conducted experiments can be observed in the following table VI.

Model	SC	CH Inx	DB Inx
DBSCAN1	-0.6250	0.0256	1.6699
DBSCAN2	-0.6931	0.0463	1.5928
DBSCAN3	-0.7387	0.0264	1.6175
OPTICS1	-0.3347	0.0317	1.8937
OPTICS2	**-0.4789**	**0.0602**	**1.8823**
OPTICS3	-0.6119	0.0962	1.8626
HDBSCAN1	-0.3752	0.0408	1.8681
HDBSCAN2	**-0.4661**	**0.0871**	**1.8256**
HDBSCAN3	**-0.5368**	**0.1123**	**1.8295**

TABLE VI
PERFORMANCE METRICS OF CLUSTERING ALGORITHMS ON 2-STEP SUB-GRAPHS WITH EXTENDED FEATURES.

In Table VII, we can visualize the label-based metrics of the top 3 models generated with 2-step graphs.

Model	N-steps	HC	NC	OC	Mean	Std
OPTICS1	1	257	52	823	0.4881	0.1922
OPTICS2	**2**	**742**	**76**	**31,631**	**0.6828**	**0.2557**

TABLE VII
RESULTS OF THE LABEL-BASED METRICS FOR THE BEST MODEL ON 1-STEP WITH EXTENDED FEATURES AND 2-STEPS MODEL.

C. Discussion

The results obtained offer a broad perspective on the depth of the approach maintained in this study. On one hand, we have the approach based on the number of features extracted from subgraphs, aiming to extract as much information as possible while reducing computational cost. We can observe very similar results in terms of cluster formation metrics, with slightly better performance in the reduced feature approach as seen in the Table IV.

However, if we look at the formation of the clusters and the metrics generated in this study to determine their quality through the labels of each of the employed directions, we can see that despite having better results in the cluster formation metrics, the results obtained in the label-based metrics for graphs with extended features show a higher number of generated clusters, with greater confidence in those clusters that are deterministic. This is relevant to the focus of the study, which aims to explore an approach to extract the behavior of subgraphs, Table V.

The results obtained appear to be clearly dependent on the depth of the subgraphs, as well as the quantity of features provided by each subgraph to achieve more precise clustering based on their unique behavior.

As we can observe in Table VI, on one hand, we again have the metrics for cluster formation, where we see much worse results compared to those obtained by 1-step depth graphs, Table IV. In the three metrics, we can highlight very low results. However, if we pay attention to the metrics that evaluate the formation of the clusters and their homogeneity, the results are much more promising.

We obtain a higher number of homogenous and determined clusters, with a greater presence of unique classes within them, offering higher values between 0.67 and 0.68 for the presence of deterministic classes in each cluster, with a standard deviation between 0.23 and 0.25 among the values, as shown in Table VII. This is offered by the models with the highest scores in the cluster formation metrics. the performance is significant compared to the values offered by normal 1-step graphs, where we have the highest performing model, DBSCAN3, Table V, in the cluster formation metrics, offering an average value of 0.28 in the predominant class and a Standard Deviation of 0.10.

Therefore, we can determine that the greater the depth and the number of extracted features, the higher the homogeneity of the clusters based on the labels of the addresses that comprise them. As can be seen in Table VII o OPTICS2 model, there is a considerable increase in the global percentage presence of the dominant class. Consequently, there is also a slight increase in the standard deviation, but with very solid numbers in the groupings of behaviors. These are grouped into a greater number of Homogeneous Clusters than those offered by 1-step graphs.

Another relevant point is the distribution of deterministic classes among these, as we have 9 different classes, it is important to know how the models not only generate clusters and their internal homogeneity but also how they distribute

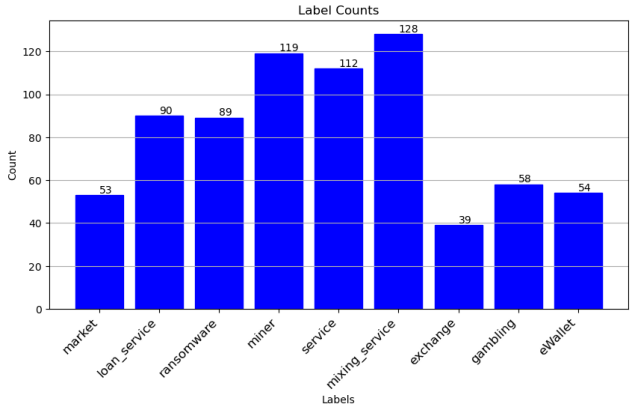

Fig. 6. Distribution of Homogeneous Clusters Across the 9 Explored Labels.

these among the different labels. In Figure 6, generated from the distribution obtained by the model with the best performance in the cluster generation metrics, OPTICS2, we achieve a fairly homogeneous distribution among the classes. We have a predominant class, mixing_service, with 128 samples, followed by a selection of classes ranging between 120 and 60 samples. Lastly, there are 3 labels, market, exchange, and eWallet with a lower sample count, this can determine a greater number of homogeneous behaviors and a higher aggregation into fewer clusters.

The distributions obtained in the different labels demonstrate that, despite having classes with a higher count, a certain balance can be observed in the complete distribution of deterministic clusters, closely related to the depth of the graphs and the number of features.

VI. Conclusion

This study provides an insight into the significance of depth and volume of graphs for analyzing the behavior of a central node that interconnects the said graph. From this perspective, several avenues are opened for consideration. The first and most relevant of these is that this approach offers a straightforward relationship between the information on the behavior of graphs based on the specified depth of relationships. It is evident that the greater the depth N of the graphs, the more the computational cost and the time required to extract and process each graph increases.

Another important aspect to highlight is the volume of data to be managed. When working with density algorithms, a larger sample space, especially at greater depths in the graphs, could provide relevant information for making more precise groupings. This also allows for the extraction of more heterogeneous information within the same cluster about different behaviors for similar samples.

In this study, three metrics were selected for evaluation: the Silhouette Coefficient (SC), the Calinski-Harabasz Index (CH Index), and the Davies-Bouldin Index (DB Index). These metrics assess the formation of clusters by different models. However, in cases of greater depth, such as two-step graphs, these metrics do not seem to provide values consistent with the quality of the clusters formed if we focus on label-based metrics, or even in the distribution of homogeneous clusters among the labels. This is the case with OPTICS2, which shows very low values in the formation metrics as seen in Table VI. However, it performs well in label-based metrics VII and in verifying the distribution across different labels as shown in Figure 6.

In future work, there is an intention to further explore this approach with a larger sample space, aiming for a greater heterogeneity of behaviors through an increased number of labeled samples. Perhaps exploring other validation metrics or models, which may or may not be density-based, could also be considered. However, with a clear focus on increasing the sample size in the case of 2-step graphs to gain a more comprehensive insight into their behavior, aiming to obtain more information on the behavior of the transactions.

ACKNOWLEDGMENTS

This work has been partially supported by the European Union's Horizon Europe programme under the project CEDAR (Grant agreement №: 101135577).

References

[1] Akcora, C.G., Li, Y., Gel, Y.R., Kantarcioglu, M.: Bitcoinheist: Topological data analysis for ransomware detection on the bitcoin blockchain (2019)
[2] Alarab, I., Prakoonwit, S.: Graph-based lstm for anti-money laundering: Experimenting temporal graph convolutional network with bitcoin data. Neural Processing Letters **55**, 689–707 (2023). https://doi.org/10.1007/s11063-022-10904-8, accepted: 25 May 2022, Published: 16 June 2022, Issue Date: February 2023
[3] Ampel, B., Otto, K., Samtani, S.: Disrupting ransomware actors on the bitcoin blockchain: A graph embedding approach (10 2023)
[4] Ankerst, M., Breunig, M.M., Kriegel, H.P., Sander, J.: Optics: Ordering points to identify the clustering structure. SIGMOD Rec. **28**(2), 49–60 (jun 1999). https://doi.org/10.1145/304181.304187
[5] Baumann, A., Fabian, B., Lischke, M.: Exploring the bitcoin network. In: 10th International Conference on Web Information Systems and Technologies (WEBIST). Institute of Information Systems, Humboldt University Berlin, Barcelona, Spain (2014), spandauer Str. 1, 10178 Berlin, Germany
[6] Bhattacharjee, P., Mitra, P.: A survey of density based clustering algorithms. Frontiers of Computer Science **15**, 1–27 (2021)
[7] Boldi, P., Vigna, S.: Axioms for centrality (2013)
[8] Caliński, T., Harabasz, J.: A dendrite method for cluster analysis. Communications in Statistics-theory and Methods **3**(1), 1–27 (1974)
[9] Campello, R.J.G.B., Moulavi, D., Sander, J.: Density-based clustering based on hierarchical density estimates. In: Pei, J., Tseng, V.S., Cao, L., Motoda, H., Xu, G. (eds.) Advances in Knowledge Discovery and Data Mining. pp. 160–172. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
[10] Chaudhari, D., Agarwal, R., Shukla, S.K.: Towards malicious address identification in bitcoin. In: 2021 IEEE International Conference on Blockchain (Blockchain). pp. 425–432. IEEE (2021)
[11] Davies, D.L., Bouldin, D.W.: A cluster separation measure. IEEE transactions on pattern analysis and machine intelligence (2), 224–227 (1979)
[12] Ester, M., Kriegel, H.P., Sander, J., Xu, X., et al.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: kdd. vol. 96, pp. 226–231 (1996)
[13] Fleder, M., Kester, M.S., Pillai, S.: Bitcoin transaction graph analysis (2015)

[14] Gaihre, A., Luo, Y., Liu, H.: Do bitcoin users really care about anonymity? an analysis of the bitcoin transaction graph. In: 2018 IEEE International Conference on Big Data (Big Data). pp. 1198–1207 (2018). https://doi.org/10.1109/BigData.2018.8622442
[15] Greaves, A., Au, B.: Using the bitcoin transaction graph to predict the price of bitcoin. In: Unknown Conference. Stanford (2015)
[16] Huang, Z., Huang, Y., Qian, P., Chen, J., He, Q.: Demystifying bitcoin address behavior via graph neural networks (2022)
[17] Magnien, C., Latapy, M., Habib, M.: Fast computation of empirically tight bounds for the diameter of massive graphs (2009)
[18] Moradi, P., Rostami, M.: A graph theoretic approach for unsupervised feature selection. Engineering Applications of Artificial Intelligence **44**, 33–45 (2015). https://doi.org/https://doi.org/10.1016/j.engappai.2015.05.005
[19] Park, S., Oh, S., Kim, H.: Performance analysis of dag-based cryptocurrency. In: 2019 IEEE International Conference on Communications Workshops (ICC Workshops). pp. 1–6 (2019). https://doi.org/10.1109/ICCW.2019.8756973
[20] Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of computational and applied mathematics **20**, 53–65 (1987)
[21] Saxena, R., Arora, D., Nagar, V.: Efficient blockchain addresses classification through cascading ensemble learning approach. International Journal of Electronic Security and Digital Forensics **15**(2), 195–210 (2023)
[22] Shah, R.S., Bhatia, A., Gandhi, A., Mathur, S.: Bitcoin data analytics: Scalable techniques for transaction clustering and embedding generation. In: 2021 International Conference on COMmunication Systems and NETworkS (COMSNETS). pp. 1–6 (2021). https://doi.org/10.1109/COMSNETS51098.2021.9352922
[23] Tharani, J.S., Charles, E.Y.A., Hóu, Z., Palaniswami, M., Muthukkumarasamy, V.: Graph based visualisation techniques for analysis of blockchain transactions. In: 2021 IEEE 46th Conference on Local Computer Networks (LCN). pp. 427–430 (2021). https://doi.org/10.1109/LCN52139.2021.9524878
[24] Weber, M., Domeniconi, G., Chen, J., Weidele, D.K.I., Bellei, C., Robinson, T., Leiserson, C.E.: Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics (2019)
[25] Zhang, P., Wang, J., Li, X., Li, M., Di, Z., Fan, Y.: Clustering coefficient and community structure of bipartite networks. Physica A: Statistical Mechanics and its Applications **387**(27), 6869–6875 (dec 2008). https://doi.org/10.1016/j.physa.2008.09.006
[26] Zola, F., Bruse, J., Eguimendia, M., Galar, M., Orduna, R.: Bitcoin and cybersecurity: Temporal dissection of blockchain data to unveil changes in entity behavioral patterns. Applied Sciences **9**, 5003 (11 2019)
[27] Zola, F., Eguimendia, M., Bruse, J.L., Urrutia, R.O.: Cascading machine learning to attack bitcoin anonymity. In: 2019 IEEE International Conference on Blockchain (Blockchain). pp. 10–17. IEEE (2019)
[28] Zola, F., Segurola, L., Bruse, J.L., Galar, M.: Temporal graph-based approach for behavioural entity classification. In: Investigació n en Ciberseguridad. Ediciones de la Universidad de Castilla-La Mancha (2021). https://doi.org/10.18239/jornadas$_2$021.34.12