Identifying Suspicious Blockchain Transactions using Clustering with Explainability

JEYAKUMAR SAMANTHA THARANI[*], Griffith University, Australia and University of Jaffna, Sri Lanka
EUGENE YUGARAJAH ANDREW CHARLES[†], University of Jaffna, Sri Lanka
PUNIT RATHORE[‡], Indian Institute of Science, India
ZHÉ HÓU[*], Griffith University, Australia
MARIMUTHU PALANISWAMI[§], University of Melbourne, Australia
VALLIPURAM MUTHUKKUMARASAMY[*], Griffith University, Australia

Blockchain is a distributed ledger technology that provides pseudo-anonymity among participants to maintain privacy. However, malicious actors utilise this property to hide their illegal rewards received through cyber attacks, dark market trades, money laundering, and Ponzi schemes. The recent confiscation by the FBI of more than $4 million USD worth of bitcoin from the 'Silk Road' dark marketplace indicates the scale of the problem faced by financial regulators and law enforcement authorities. Analysing and identifying harmful actors is, therefore, necessary to regulate the transactions of digital assets. Machine learning models can assist in detecting patterns and correlations between the actors in blockchain networks that may not be apparent through traditional methods. In blockchain networks, the number of actors linked to illegal activities is significantly smaller than that of regular activities. Also, only very limited labelled transaction data is available about these malicious actors. These limitations make it harder to train supervised learning models to provide real-time proactive responses. This paper represents a pioneering effort in thoroughly examining the different unsupervised learning methods for clustering suspicious behaviour of actors within blockchain networks. The proposed unsupervised learning-based analysis considers metadata and interconnectivity information of blockchain transactions. The metadata contains time-based and amount-based information. Interconnectivity data represents centrality measures and embedding vectors of the blockchain network. The quality of the identified clusters is validated using internal and external cluster validation measures. The validation results were used to identify influential features using the eXplainable AI technique Shapley (ShAP) values. The results reveal that the features related to the spending and receiving transactions strongly influenced cluster identification. Overall, the centroid-based and connectivity-based approaches identified well-separated clusters for metadata and centrality-based features of blockchain transactions.

CCS Concepts: • **Blockchain network** → Financial activities; • **Ransomware settlement** → **Ponzi smart contract**; • **Clustering**;

Additional Key Words and Phrases: blockchain, unsupervised learning, ransomware, Ponzi scheme, and community detection.

ACM Reference Format:
Jeyakumar Samantha Tharani, Eugene Yugarajah Andrew Charles, Punit Rathore, Zhé Hóu, Marimuthu Palaniswami, and Vallipuram Muthukkumarasamy. 2018. Identifying Suspicious Blockchain Transactions using Clustering with Explainability. In *Proceedings of*

Authors' Contact Information: Jeyakumar Samantha Tharani, samantha@univ.jfn.ac.lk, Griffith University, Australia and University of Jaffna, Sri Lanka; Eugene Yugarajah Andrew Charles, University of Jaffna, Sri Lanka; Punit Rathore, Indian Institute of Science, India; Zhé Hóu, Griffith University, Australia; Marimuthu Palaniswami, University of Melbourne, Australia; Vallipuram Muthukkumarasamy, Griffith University, Australia.

Manuscript submitted to ACM

Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX). ACM, New York, NY, USA, 22 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Blockchain is a decentralised digital ledger that records transactions across multiple nodes (computers) connected in a peer-to-peer manner to achieve transparency and a tamper-proof system [31, 44]. The major components of this decentralised network are structured using blocks, transactions, hash pointers, wallets, and smart contracts. A block contains a set of transactions corresponding with the nodes' wallets or smart contracts [14, 40]. These blocks are connected chronologically using the hash of block information and then formed as a blockchain. Pseudo-anonymity is an essential blockchain property that helps maintain privacy by obscuring the real-world identities of the nodes using cryptographic techniques [5]. The exploitation of the pseudo-anonymity aspect by malicious actors has become a concern within blockchain applications. These malicious actors may leverage the privacy features of blockchain to engage in malicious activities, such as money laundering, ransomware, Ponzi scheme payments, and dark market trades [13]. The data relating to illegal activities are known as anomalies, often categorised into point, collective and contextual [10]. A point anomaly is a data point that deviates from the expected pattern, range, or norm. A collective anomaly is a data point that may not be considered an anomaly by itself, but the occurrence of multiple points indicates an anomaly. A contextual anomaly is a data point of anomaly viewed against meta-information associated with the data points. The suspicious activities often associated with blockchain networks combine collective and contextual categories [17]. Each collective and contextual category has distinctive features that become input for the study related to suspicious behaviour identification in blockchain networks. In such analyses, supervised and unsupervised learning-based approaches play crucial roles. Supervised learning emphasises prediction accuracy based on the label of data, and unsupervised learning focuses on uncovering novel patterns and relationships in data.

Supervised learning is 1) subjective for the given labels [26, 38, 41, 42], 2) model training needs a large amount of labelled data, which is a challenging part of blockchain analysis 3) not able to handle categorical or correlated attributes, which can be common in the transactions of the blockchain network, and 4) may not generalise well to new, unseen data, leading to overfitting or poor performance on validation datasets, which is the real scenario in the streaming blockchain transaction data. However, unsupervised learning does not require label information to train the model; it may handle imbalanced data [8], and these models can interpret unknown anomalies in the data as a new group. Thus, this research examines various unsupervised approaches to identify groups of malicious actors in blockchain networks.

Current literature on blockchain analysis typically uses unsupervised learning to visualise various behaviours of blockchain transaction data. The research works proposed by Monamo et al., [25], Pham et al., [29], and Martin et al., [23] performed unsupervised learning using k-means to visualise abnormal behaviours of the Bitcoin transaction based on currency, network, graph embedding, and time-series-based features. Other studies conducted by Zhang et al., [46], and Huang et al., [16] proposed clustering algorithms *BTCOut*, and *BPC*, respectively, to analyse the anomalous behaviour of the users and transactions in the Bitcoin network. Podgorelec [30] proposed an unsupervised learning-based anomaly detection approach to improve the usability of decentralised applications (DAPPs). Their experiment used public Ethereum time series-based transaction data and clustered them using the Isolation Forest approach.

The literature review highlights the state-of-the-art unsupervised methods to analyse blockchain transaction clustering for detecting abnormal behaviour within specific networks. However, the quality of identified clusters remains unvalidated, and the influence of transaction features on cluster identification is not analysed. Furthermore, the research on graph embedding-based clustering in [23] lacks validation against blockchain data. Considering these limitations, this research study exploited the Bitcoin and Ethereum transactions with their metadata and behavioural information-based features. Five different clustering approaches are considered to determine the most appropriate methods for cluster identification within the blockchain network. The optimal clustering approaches were determined through internal and external evaluation metrics. The best clustering approach is then used to identify the features that influence cluster identification.

The contributions of this research are:

(1) This research addressed the gap in exploiting the intersection of the security of blockchain transactions and the best use of unsupervised learning approaches and explainable AI. This study employed unsupervised learning techniques, using actual blockchain transaction metadata, to overcome the limitations of supervised learning-based real-time analysis. The results showed that centroid-based and connectivity-based methods can efficiently identify the clusters in blockchain networks.

(2) This research identified the most effective clustering techniques for blockchain transaction analysis. The study evaluated the effectiveness of identified clusters for three categories of features: structural, network property-based, and graph-embedding, using internal and external measures. This is useful for implementing a robust clustering approach to support real-time decision-making regarding the suspicious behaviour of actors in blockchain networks.

(3) This research examined the underlying differences in the generated clusters by analysing the probability distributions of the influential features. The study adopted a novel approach to explain cluster identification; to the best of our knowledge, it is the first of its kind. This study analysed influential feature identification of wallet and smart contract transactions using the explainable AI technique Shapely values. The probability distribution of the feature outcomes provided explanations for the identified clusters contributing to the theoretical development of unsupervised learning models for analysing blockchain networks.

The findings of this work may lead to the implementation of an unsupervised learning model for analysing blockchain transactions in real time to improve the security of the blockchain network. The structure of the paper is as follows: Section 2 presents a critical review of state-of-the-art techniques. Section 3 details the proposed methodology. Section 4 describes the data collection, feature extraction and pre-processing. Section 5 explains the experimental setup and performance analysis, and Section 6 discusses the findings based on the cluster evaluation scores and the feature importance outcome of the eXplainable AI technique. Section 7 concludes with the research findings and limitations.

2 RELATED WORK

This section critically examines relevant literature on unsupervised learning in analysing blockchain transactions. The related work used clustering and non-clustering approaches to analyse and visualise the behaviour of the transactions in the Bitcoin network. The following related works illustrate the contributions using clustering approaches to detect anomalies or illicit activity in cryptocurrency networks, specifically focusing on Bitcoin transaction data.

Clustering approaches: Monamo et al. [25] conducted unsupervised cybercrime detection in the Bitcoin network using the trimmed k-means clustering approach. However, the identified clusters lacked validation and clarity regarding the features employed for the clustering process. The approach does not clarify how these clusters are associated with cybercrime within the Bitcoin network.

Wahrstatter et al., [39] utilised the unsupervised learning approach k-means to analyse the Bitcoin user graph to identify the suspicious actors dealing with illicit activities. Their study introduced four distinct categories of features:

turnover, connectivity, activity, and UTXO-specific features. Evaluation of these proposed features was undertaken through a comparison of the distinguished clusters. The data used in their research was based on illicit activities, and the clusters were validated manually using the ground truth labels. However, their proposed approach is limited to Bitcoin transactions related to CoinJoin wallets.

Rui Zhang et al. [46] proposed a clustering approach *BTCOut* based on meta path, which captures the various relations and constraints in the Bitcoin network to address the challenge of identifying abnormal behaviours or transactions within the network. Their approach considers user-to-user and transaction-to-transaction interactions within the Bitcoin network to determine the candidate set of target nodes and fusion similarity scores. The recall, precision, and F2-score of *BTCOut* outperformed baseline algorithms such as k-medoids, DBSCAN, OddBall, CTOutliers, and Tclust.

Butian et al. [16] proposed a novel behavioural pattern clustering (BPC) algorithm to cluster nodes in blockchain networks based on generated sequences. The sequences are changed based on the number of transactions over time. These sequences are then used to calculate the dynamic time warping (DTW) similarity scores, and then the BPC algorithm leverages these similarity scores to cluster the nodes. Their proposed approach outperformed well-known clustering methods such as DBSCAN and HIC (Hierarchical Clustering Methods).

Kevin et al. [23] conducted research utilising multiple machine-learning models to detect anomalous transactions in various digital currency networks. Their study explored the effectiveness of supervised and unsupervised approaches for detecting anomalous nodes. In the unsupervised approach, first, they employed the GAT2VEC embedding algorithm to extract features from the network. Then these embedding features were combined with the degree, currency, and time-based features. The resulting combined feature set was then used as input for the k-means clustering approach. However, it should be noted that the data used for the experiments and the results mentioned in their research were not based on blockchain networks.

The following literature summarises studies that utilised non-clustering approaches with clustering approaches to detect abnormal behaviours and patterns within blockchain networks.

Non-clustering approaches: Pham et al. [29] investigated the abnormal behaviour of Bitcoin transactions using two network analysis approaches, such as power degree and densification laws, an unsupervised learning-based approach, k-means clustering, and an outlier detector non-clustering approach, Local Outlier Factor (LOF) [7]. The network analysis approaches utilised six features of the user graph and three features of the transaction graph. These features are based on the number of input (in-degree) and output (out-degree) transactions and their total and mean incoming and outgoing transaction amounts. The total transaction amount feature revealed that the anomalous behaviour for power degree distribution values, and the mean of the incoming and outgoing transaction amount features, support identifying anomalies using the LOF approach. No specific findings have been reported for the k-means clustering.

Priyanshi et al. [36] conducted anomaly detection in a blockchain network using transaction data from the Bitcoin network. First, their proposed approach selects features by applying sequential forward feature selection, which was then used as inputs for various machine learning approaches. They utilised non-clustering approaches, SVM, isolation forest, and clustering approach, k-means algorithm for the analysis and achieved the highest accuracy with the SVM classifier. However, the research lacks information regarding the specific value of k used in the k-means algorithm, cluster validation, and features considered for clustering.

Podgorelec et al., [30] used a non-clustering Isolation Forest [21] to isolate each transaction and split them into inliers (normal transactions) and outliers (anomalous transactions), based on the number of decisions in the decision tree to isolate the transaction to visualise abnormal behaviour via distinguished clusters of the smart contract and

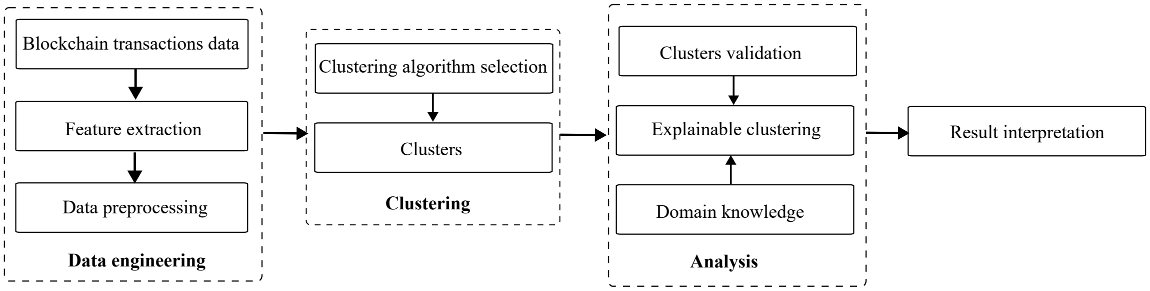


Fig. 1. Overview of the proposed methodology.

Externally Owned Accounts (EOAs) of Ethereum main network within a given timeframe. However, identified clusters were not validated, and the smart contract features used for clustering were not reported.

Overall, state-of-the-art unsupervised analyses of transactions are performed and visualised using only specific blockchain networks. The identified resultant clusters based on various approaches were not validated. Furthermore, the influence of the metadata and interconnectivity-based features in clustering was not analysed. Considering these limitations, this study is novel. (i) identified suitable clustering approaches for blockchain analysis via validated outcomes of various unsupervised learning approaches. The validation considers internal and external evaluation measures that enhance the credibility and reliability of clustering analysis, and (ii) the significance of the metadata and interconnectivity information-based features of Bitcoin transactions and Ethereum smart contracts analysed using XAI to identify the most relevant features for clustering transactions.

Performing clustering analysis in real-time may provide timely detection and response for financial regulators, law enforcement authorities, and forensic analysers, regarding the emerging patterns or anomalies in transaction data. This requires the development of efficient algorithms and scalable approaches. The contributions of this study laid a foundation for the implementation of an incremental clustering approach for blockchain real-time data analysis.

3 OVERVIEW OF THE RESEARCH METHODOLOGY

This section outlines the research methodology for unsupervised learning in financial crime-related blockchain transactions. The research methodology involves three major phases as shown in Fig. 1 such as (1) data engineering - collecting data from public blockchains and extracting features from transactions and preparing them for clustering, (2) clustering transaction data using nine unsupervised learning approaches and validating using internal and external evaluation measures, and (3) feature importance analysis to identify most relevant features for distinguishing between transactions.

Data engineering includes data collection, feature extraction, and data preprocessing. First, the data collection considers Bitcoin and Ethereum's normal and crime-related transactions. The dataset for Bitcoin considers transactions corresponding with the wallets that deal with normal and ransomware settlements. The labels used for cluster validation indicate whether the group of transactions is normal or related to ransomware settlements. The dataset for Ethereum considers normal and Ponzi schemes-based smart contract transactions, and labels for validation indicate whether the group of smart contracts are Ponzi schemes or not. Then feature extraction extracts three categories of feature sets: structural, network property-based, and embedding. The structural feature set includes the fields in the transaction metadata and their statistical measuress [38]. Network property-based features use centrality measures [32] and embedding features utilised random walk-based embedding framework GraphSAGE [15], on transaction graph and hypergraph of blockchain networks [37]. Finally, data preprocessing normalises the values of the engineered features within the range of [0, 1] and prepares them for clustering.

The clustering phase uses k-means to set the baseline for unsupervised learning and other state-of-the-art approaches, such as Density-Based Spatial Clustering of Applications with Noise (DBSCAN), spectral clustering, affinity propagation (AP), Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH), Ordering Points To Identify the Clustering Structure (OPTICS), mean-shift, Gaussian Mixture Model Clustering (GMM), and Deep Embedding Clustering (DEC). The clustering approach considers each category of feature set separately and obtains the clusters which indicate the groups of transactions or smart contracts.

The analysis phase involves three tasks: cluster validation, explainable clustering, and verification using domain knowledge. The cluster validation provides scores for identified clusters using internal and external evaluation measures. Based on the validation score, the best clustering approach was selected for each category of features. Cluster outcomes obtained from the best approaches are considered for explainable clustering. The explainable clustering justifies cluster identification using the eXplainable AI technique Shapley values [22]. The Shapley value is an effective measure to identify the influence of the features in cluster outcomes. The outcome of the explainable clustering is compared with the domain knowledge, such as the behaviour of the financial crime and the nature of the clustering approach.

Finally, the result interpretation considers the analysis outcomes to (1) identify suitable clustering approaches for different categories of features, (2) analyse the influence of the features (structural or graph-based) in cluster identification, and (3) determine the feasibility of unsupervised learning approaches in realtime blockchain network analysis. The following sections detail the implementation processes of each phase involved in the proposed approach.

4 DATA ENGINEERING

This section describes data collection from public Bitcoin and Ethereum blockchains, feature extraction, and data preprocessing.

4.1 Data collection

The data collection considers normal and ransomware settlement-based transactions of Bitcoin networks and non-Ponzi and Ponzi smart contract transactions of Ethereum networks.

Normal and Ransomware settlements' transactions: The normal and ransomware settlement-related Bitcoin transactions are downloaded using public API [1]. The data sample contains 21,678 ransomware transactions and 33,069 normal transactions. The labels of the transactions referenced from the public dataset BitcoinHeist [1] dataset.

Normal and Ponzi smart contract transactions: The non-Ponzi and Ponzi smart contract transactions are collected using public API [2]. The collected dataset contains 200 Ponzi and 3,590 non-Ponzi smart contract transactions. The labelled information of the transactions was obtained from the public datasource [3].

4.2 Feature extraction

Feature extraction involves generating features under three categories for transaction data described in subsection 4.1. The three categories are structural, network property-based, and embedding features. The structural features are

[1] https://www.blockchain.com/api/blockchain_api
[2] https://www.blockcypher.com/
[3] https://www.kaggle.com/datasets/xblock/smart-ponzi-scheme-labels

derived from metadata information on Bitcoin transactions. The primary elements of the Bitcoin data are incoming and outgoing transactions, timestamps, and transaction fees. The rest of the features are statistical measures(maximum, minimum, mean, median, and standard deviation) of these elements. Table 1 lists details of the Bitcoin transaction's structural features; the features $inout - ratio$ and $unique - out$ are adopted from the research work reported in [19], and described as in equation (1). The other transaction features are adopted from the research work reported in [38].

$$\textit{inout-ratio} = \frac{n(\mathcal{I})}{n(O)} \tag{1}$$

Here $\mathcal{I}$ represents a set of input transactions and O represents a set of output transactions. The $n(\mathcal{I})$ and $n(O)$ are the total number of input and output transactions, respectively.

The major components of the smart contract are the spending or receiving amount, timestamp, and amount of gas. The smart contract receives income when invoked by another smart contract or wallet, whereas it is spent when invoking another smart contract. The structural features of the Ponzi smart contracts are derived using the information available in contract creation and invocation transactions and their timestamps as described in Table 2. These structural features are described in equations (2), (3), (4), and (5) using an approach similar to the one used in [38] to extract EOA features.

$$\textit{asSender} = f(u_x), \text{ where } u_x \in S \tag{2}$$

$$\textit{asReceiver} = f(u_x), \text{ where } u_x \in R \tag{3}$$

$$\textit{totalSpent}(u_x) = \sum_{i=1}^{m} |t_{u_x v_i}|, \text{ where } v_i \in R \tag{4}$$

$$\textit{totalReceive}(u_x) = \sum_{i=1}^{n} |t_{v_i u_x}|, \text{ where } v_i \in S \tag{5}$$

Where $f(u_x)$ represents the number of times smart contract u_x participated as a sender or receiver, R is the list of transactions invoked by smart contract u_x (receiving), S is the list of transactions invoked by the smart contract u_x (spending), and $|t_{u_x v_i}|$ is the amount spent or received by the smart contract u_x in transaction v_i. Here, n and m are the total number of spending and receiving transactions, respectively.

The network property-based and graph embedding features are extracted from the transaction graph and hypergraph of blockchain networks. The transaction graph G_t is constructed using input and output elements of Bitcoin transactions. In this graph, each node is a transaction. The edge information $e(u, v)$ indicates the Unspect Transaction Output (UTXO) transferred from transaction u to v. The hypergraph G_h, which is a generalised graph, can be utilised to represent blockchain transactions generated by Bitcoin or Ethereum networks. G_h represents the interactions between a transaction u and a wallet or smart contract v. The edge information $e(u, v)$ indicates the value of cryptocurrency received or spent by transaction and confirmation time [18]. The transaction data related to the account-based transaction model does not have connections between transactions, as in the UTXO model, since this study only considered a hypergraph for the Ethereum network.

The network property-based features of Bitcoin transactions and Ethereum smart contracts in Table 3 and Table 4 were generated based on the degree, betweenness, closeness, and eigenvector centrality measures of the actors (wallets/smart contracts/transactions) in blockchain networks [38]. In Table 3, the input indicates a wallet receives Bitcoins on the transaction, and the output represents the wallet spends Bitcoins from the transaction.

Finally, graph embedding features $\mathcal{H}_u$ in equation (6) are extracted using the GraphSAGE [15] approach by considering topology $\mathcal{T}$ of transaction graph G_t or hypergraph G_h [18].

$$\mathcal{H}_u = f(\mathcal{T}, \mathfrak{R}(u, v), \mathcal{I}) \tag{6}$$

Here, $\mathfrak{R}(u, v)$ represents the spending or receiving relation between the transaction-to-wallet or transaction-to-transaction of the Bitcoin and the transaction-to-smart contract of the Ethereum. Node properties $\mathcal{I}$ represent the transactions and smart contracts' structural properties presented in Table 1 and Table 2, respectively. The dimension of $\mathcal{H}_u$ is 64.

Table 1. Structural features for Bitcoin transactions.

Feature	**Description**
$inDegree$	number of incoming transactions
$outDegree$	number of outgoing transactions
$totalInput$	total amount of Bitcoins received
$totalOutput$	total amount of Bitcoins sent
$(*) - input$	(min/ avg/ max/ med/ mod/ std) amount of Bitcoins received
$(*) - output$	(min/avg/max/med/mod/std) amount of Bitcoins spent
$inout - ratio$	ratio between the number of inputs and outputs
$unique - out$	number of unique output addresses involved in a transaction

4.3 Data preprocessing

In data preprocessing, values of each feature of transactions and smart contracts are normalised between [0,1]. The dataset for Bitcoin transactions does not have any null or categorical values. The smart contract data entries labelled as 'error' are removed from the dataset.

5 CLUSTERING AND VALIDATION

This section details clustering-based analysis for Bitcoin and Ethereum networks using their transactions and three categories of features. The clustering outcomes were then validated using internal and external evaluation measures. The evaluation scores were used to select suitable clustering approaches for each category of features based on silhouette scores and F1-score values. The cluster IDs corresponding to the best-performing unsupervised learning approaches are considered for feature importance analysis using the eXplainable AI technique and domain knowledge. The results of feature importance provide insights into the significance of each category of features in distinguishing clusters.

5.1 Experimental setup for clustering

The experimental setup selected 54,747 samples of Bitcoin transactions and 3,790 samples of smart contract transactions from the datasets stated in sub-section 4.1, then preprocessed. The preprocessed samples were grouped utilising clustering approaches on four different aspects: connectivity, centroid, density, and distribution; the details are stated below.

Table 2. Structural features for smart contracts.

Feature	Description
$asSender$	number of times a smart contract as a sender
$asReceiver$	number of times a smart contract as a receiver
$totalSpent$	total amount spent by a smart contract
$totalReceive$	total amount received by a smart contract
$(*) - spent$	(min/avg/max/med/mod/std) amount spent by a smart contract
$(*) - receive$	(min/avg/max/med/mod/std) the amount received by a smart contract

Table 3. Network property-based features for Bitcoin transactions.

Feature	Description
$betweenness - t$	betweenness centrality value between the transactions
$betweenness - i$	betweenness centrality value between the transaction and the input wallet
$betweenness - o$	betweenness centrality value between the transaction and the output wallet
$closeness - t$	closeness centrality value between the transactions
$closeness - i$	closeness centrality value between the transaction and the input wallet
$closeness - o$	closeness centrality value between the transaction and the output wallet
$degree - t$	degree centrality value between the transactions
$degree - i$	degree centrality value between the transaction and the input wallet
$degree - o$	degree centrality value between the transaction and the output wallet
$eigenvector - t$	eigenvector centrality value between the transactions
$eigenvector - i$	eigenvector centrality value between the transaction and the input wallet
$eigenvector - o$	eigenvector centrality value between the transaction and the output wallet

Table 4. Network property-based features for Ponzi smart contracts.

Feature	Description
$betweenness$	betweenness centrality value between the smart contract and transactions
$closeness$	closeness centrality value between the smart contract and transactions
$degree$	degree centrality value between the smart contract and transactions
$eigenvector$	eigenvector centrality value between the smart contract and transactions

5.1.1 Connectivity-based clustering (Hierarchical). Connectivity-based clustering groups objects based on their proximity distance. The resulting clusters are represented as dendrograms. BIRCH [28] is a well-known hierarchical clustering approach designed to handle streaming and large-volume data points. It can identify clusters with varying shapes, densities, and sizes and is less dependent on user-defined parameters. The nature of the blockchain data is large in volume, and the clustering approach chosen for the experiment may have the potential to detect the community on streaming blockchain data. Hence, BIRCH is chosen as one of the clustering approaches.

5.1.2 Centroids-based clustering. Centroid-based clustering is the simplest of all clustering types and operates based on the proximity of data points to a chosen central value. Initially, the dataset is partitioned into a predetermined number of clusters associated with a vector of values. This vector, known as the cluster's centroid, is calculated as the average of all data points within a particular cluster. Subsequently, the data points are reorganised by considering the distance between each data point and the centroid vector. This research considered k-means, spectral clustering, and affinity propagation (AP) approaches under this category. The k-means[20] is a simple clustering approach that can provide intuitive cluster assignments for blockchain data. Whereas spectral clustering[27] can handle non-linearly separable data and effectively identify clusters with complex structures or shapes of the blockchain data. AP [45] does not require specifying the number of clusters in advance and can automatically determine it based on the data, which helps to identify several groups of wallets or smart contracts that may be related to money laundering, dark market trades or Ponzi schemes.

5.1.3 Density-based clustering. The density-based clustering method considers density instead of distance. In this approach, the data is clustered by regions of high-concentration objects bounded by areas of low-concentration objects. Here clusters are formed by maximal sets of connected data points. Popular density-based clustering approaches DBSCAN [12], OPTICS [3], and Mean shift [11] are considered for community detection. In blockchain data, the outliers may caused by the irregular behaviour of the wallets or smart contracts. The DBSCAN approach effectively handles datasets with uneven cluster sizes and outliers which helps to identify clusters of transactions that occur closely together in terms of their attributes. Blockchain data often exhibit varying densities of transactions over time or across various applications. The OPTICS is an extension of DBSCAN, which can handle varying densities is robust to the choice of distance parameters and is useful for analysing blockchain data where transaction densities may fluctuate. The clusters from OPTICS may provide insights into the organisation and relationships among different groups of transactions, wallets, and smart contracts within the blockchain networks. In streaming blockchain data, where the number of clusters and their characteristics may not be known in advance, The Mean shift approach can identify clusters with irregular shapes and sizes, does not require the number of clusters in advance, is resistant to noise and is adaptive to the density variations within the data. This is beneficial for identifying the trends within the blockchain data and detecting suspicious activities or common usage patterns.

5.1.4 Distribution-based clustering. Distribution-based clustering creates and groups data points based on their likelihood or probability distribution in data. Each cluster has a central point, and the higher the distance of the data point from the central point, the lesser the probability of getting included in that cluster. The Gaussian Mixture Model with Expectation–Maximization (GMM with EM) [47] is a well-known distribution-based clustering approach and captures groups with various shapes, sizes, and orientations. It can handle datasets that exhibit non-linear or non-spherical cluster structures. Blockchain data can exhibit complex distributions due to different types of transactions, interactions between various actors, and network dynamics. GMM can effectively model these distributions by representing the data as a mixture of multiple Gaussian distributions. Each Gaussian component captures a different cluster or pattern within the data. Transactions that do not fit into identified Gaussian components inform potentially suspicious activities within the blockchain network, such as money laundering, Ponzi schemes, or other illicit activities. The soft clustering nature of GMM is also advantageous for blockchain data analysis, as transactions may exhibit characteristics of multiple clusters simultaneously. Further, GMM is relatively scalable and can handle large datasets, which is important for extracting meaningful insights and patterns for analysing the vast amount of transaction data generated by blockchain networks.

5.1.5 Deep Clustering. Deep clustering approach [9] performs clustering of data using neural networks. It combines deep learning models with traditional clustering algorithms to extract meaningful representations and groups of similar instances. Deep Embedding Clustering (DEC) [43] is one of the deep clustering approaches. Unlike traditional clustering algorithms, DEC first converts the given data points X in a lower-dimensional feature space Z and then optimizes a clustering objective in Z. Stochastic Gradient Descent (SGD) via backpropagation on clustering is used to learn the mapping parameterised by a deep neural network. This approach iteratively refines clusters with an auxiliary target distribution derived from the current soft cluster assignment. This process gradually improves the clustering as well as the feature representation. Blockchain data typically consists of various features based on metadata and behavioural information. DEC can automatically learn meaningful representations (embedding) of these features and capture complex relationships and patterns that may not be evident in the raw feature space. Also, the embedding identified by DEC captures semantic similarities between data points, leading to more accurate and interpretable clustering results.

Unsupervised learning on blockchain transaction data is a relatively new area of research and the cluster formation depends on different parameters. Due to these reasons, this research chooses clustering approaches in each category with various perspectives. Table 5 states the configuration values used to customise the clustering approaches described above. The number of clusters, k, for k-means, n for spectral clustering, and c for GMM were obtained using the elbow method [35], the optimal ϵ value for DBSCAN and OPTICS, threshold, t, value for BIRCH, and bandwidth, b, for Mean shift are identified by calculating the distance between data point and its neighbour in cluster. The distances were then sorted in descending order to identify the knee in the elbow method [6]. The value corresponding with the knee is considered optimal for the ϵ or b. In Table 5, the columns *Structural*, *Network*, and *Embedding* list the parameter values identified by the incorporation of structural, network property-based, and graph embedding features. In this table, the ‘Trans‘ and ‘Hyper‘ columns in *Embedding* indicate the parameter values obtained for the transaction graph and hypergraph embedding features, respectively. The following sub-section presents unsupervised learning results for each eight clustering approaches by using three categories of features.

Table 5. Parameter values for selected clustering approaches. Here, ’Trans‘ and ‘Hyper‘ columns indicate the embedding features obtained using the transaction graph and hypergraph, respectively.

Approach	**Parameter**	**Configuration**						
		Bitcoin				Ethereum		
		Structural	Network	Embedding		Structural	Network	Embedding
				Trans	Hyper			Hyper
k-means	k	2	2	2	2	2	2	2
DBSCAN	ϵ	0.7705	0.4043	0.0138	0.0312	0.48006	0.11031	0.07658
Spectral	n	2	2	2	2	2	2	2
AP	d	0.9357	0.9357	0.9357	0.9356	0.9355	0.9355	0.9355
BIRCH	t	0.9	0.1	0.5	0.8	0.8	0.2	0.7
OPTICS	ϵ	0.9909	0.8252	2.8927	0.0736	0.59789	0.56298	0.8971
Mean shift	b	0.7705	0.4043	0.0318	0.0312	0.48006	0.11031	0.07658
GMM	c	2	2	2	2	2	2	2

5.2 Validation

The first phase of analysis validates the performance of examined clustering approaches using an internal validation measure, Silhouette Coefficient (SC) [34], and three external validation measures, Adjusted Rand Index (ARI) [33], Normalised Mutual Information (NMI) [2], and F1-score. By considering the need for real-time blockchain transaction analysis, this experiment validates the clusters using internal validation measures. Moreover, external validation measures are used to understand the meaning of the identified clusters using the correlation between actual and cluster IDs. These measures not only help in measuring performance but also go a long way in helping to understand the relation between the features and cluster identification.

5.2.1 Internal validation. The internal validation matrices [4] mainly depend on two measures cohesion within each cluster and separation between different clusters. The values of cohesion and separation are combined to estimate the validation score. Let S be a solution having n number of clusters $(C_1, C_2, C_3, ..., C_n)$, the validity of S will be computed as described in equation (7).

$$\text{Validity(S)} = \sum_{k=1}^{n} w_k * Validity(C_k) \tag{7}$$

Here, w_k is a weight assigned to the validity of each cluster and $Validity(C_k)$ is the validity of an individual cluster.

The cohesion for a cluster C_k detailed in equations (8) can be computed by summing the similarity between each pair of records (x, y) contained in that cluster.

$$\text{Cohesion}(C_k) = \sum_{x \in C_k; y \in C_k} \text{Similarity}(x, y) \tag{8}$$

The separation between two clusters C_j and C_k stated in equation (9) can be computed by summating the distance between each pair of records falling within the two clusters, and both records are from different clusters.

$$\text{Separation}(C_j, C_k) = \sum_{x \in C_j; y \in C_k} \text{Distance}(x, y) \tag{9}$$

The set of clusters having high cohesion within the clusters and high separation between the clusters is considered to be efficient. The Silhouette coefficient (SC), Dunn index, and Davies–Bouldin index are well-known internal validation approaches. Blockchain data, based on various categories of features exhibiting different patterns, clusters may appear convex or non-convex. Unlike other internal evaluation metrics, the SC can handle both convex and non-convex clusters and is not restricted to a particular clustering approach. Due to these advantages, this analysis considered SC [34] as the internal evaluation measure.

- **Silhouette Coefficient (SC):** The Silhouette coefficient method evaluates any clustering model without the actual labels and produces a score in the range between -1 and 1. A larger score indicates samples are closer to their cluster than they are to other clusters. The evaluation score is calculated by combining two subscores: the mean distance between a sample and all other points in the same cluster(α) and the mean distance between the sample and all other points in the next nearest cluster (β). Based on α and β, the Silhouette coefficient σ can be defined as in equation (10).

$$\sigma = \frac{\beta - \alpha}{max(\beta, \alpha)} \tag{10}$$

5.2.2 External validation. The external validation technique evaluates the similarity between generated and original clusters (based on actual labels). This type of validation can be carried out if actual labels are available. The cluster validation process considers these evaluation measures to analyse the accuracy of the actual labels and the correctness of clustering approaches.

In this approach, the clustering algorithm has resulted in a set of clusters $\hat{Y} = (\hat{y}_1, \hat{y}_2, \hat{y}_3, ..., \hat{y}_k)$ and clusters $Y = (y_1, y_2, y_3, ..., y_m)$ representing the actual clusters based on the labels provided in the dataset. The ARI [33], NMI [24], and F1-scores are well-known external validation approaches.

- **Adjusted Rand Index (ARI)** compares cluster assignments by making pairwise comparisons as equation (11). It determines if the prediction is accurate based on whether instances are assigned to the same or different clusters depending on whether they belong to the same or different actual clusters.

$$ARI = \frac{\sum_{ij}\binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] / \binom{N}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] / \binom{N}{2}} \tag{11}$$

  N indicates total number of data points, n_{ij} is number of data points simultaneously in resulted clusters i and actual clusters j. The a_i is the number of data points in the resulting cluster i, and b_j is the number of data points in the actual cluster j. The value of ARI ranges from [-1, 1], and a value closer to 1 indicates higher agreement between the resulting cluster and the actual cluster. The value 0 indicates a result no better than random, and a value close to -1 indicates a complete disagreement.

- **Normalised Mutual Information (NMI)** evaluates the similarity between two clusterings of the same data. Let $\hat{Y}$ be the cluster labels assigned to data points as per clustering results (predicted labels) and let Y be the actual labels of the data points. Equation (12) details the calculation for NMI.

$$NMI = \frac{MI(Y, \hat{Y})}{\frac{1}{2}[H(Y) + H(\hat{Y})]} \tag{12}$$

  Here, the mutual information (MI) between the cluster results labels $\hat{Y}$ and the actual labels Y is defined as in equation (13).

$$MI(Y, \hat{Y}) = \sum_{i=1}^{|Y|} \sum_{j=1}^{|\hat{Y}|} P(i, j) log \frac{P(i, j)}{P(i)P^*(j)} \tag{13}$$

  Here, $P(i, j)$ is the probability of data occurring in actual cluster i and predicted cluster j, $P(i)$ is the probability of data occurring in actual cluster i and $P^*(j)$ is the probability of data occurring in predicted cluster j. The $H(Y)$ is the Entropy of actual cluster assignments, and $H(\hat{Y})$ is the Entropy of predicted cluster assignments. The $|Y|$ and $|\hat{Y}|$ indicate the size of the actual and identified clusters, respectively. The NMI value of 1 indicates that the two clusters are identical, and 0 indicates that the two clusters are entirely different. F1-score is a well-known machine learning evaluation metric that assesses the predictive skill of a model by elaborating on its class-wise performance rather than an overall performance as done by accuracy.

5.2.3 Validation results. This section provides internal and external cluster validation scores corresponding to the chosen clustering methods for every feature set as outlined in section 4.2. Tables 6, 7, 8, and 9 present the cluster validation scores for the structural, network property-based, and the graph embedding features of the Bitcoin transactions, respectively.

It can be observed from Table 6 that, for the structural features of the Bitcoin transactions, the Affinity Propagation (AP) clustering approach obtained a high Silhouette score of 0.36, ARI of 0.7255, NMI of 0.6110, and F1-score of 0.9254. For the network property-based features of the Bitcoin transactions, the AP obtained the best internal and external evaluation scores as shown in Table 7.

Table 6. Results for structural features of Bitcoin transactions.

Approach	SC	ARI	NMI	F1-score
k-means	0.35	0.0978	0.0773	0.6177
DBSCAN	0.24	0.00004	0.0001	0.4553
Spectral	0.35	0.1039	0.0840	0.6265
AP	**0.36**	**0.7225**	**0.6110**	**0.9254**
BIRCH	0.33	0.0524	0.0677	0.2121
OPTICS	-0.37	0.0899	0.1264	0.5944
Mean shift	0.24	0.1611	0.1968	0.7007
GMM	0.12	0.0011	0.0006	0.4325
DEC	0.34	0.0944	0.1052	0.1983

Table 7. Results for network property-based features of Bitcoin transactions.

Approach	SC	ARI	NMI	F1-score
k-means	0.33	0.0118	0.0164	0.3235
DBSCAN	0.13	0.0382	0.1084	0.5309
Spectral	0.33	0.0128	0.0171	0.5433
AP	**0.37**	**0.6203**	**0.5013**	**0.8941**
BIRCH	0.33	0.0102	0.0152	0.3249
OPTICS	-0.2	0.3192	0.2929	0.7685
Mean shift	0.24	0.4032	0.3031	0.8184
GMM	0.33	0.0099	0.01507	0.3251
DEC	0.2	0.0896	0.0941	0.6322

Table 8. Results for transaction graph-based embedding features of Bitcoin transactions.

Approach	SC	ARI	NMI	F1-score
k-means	**1.0**	-0.0062	0.0054	0.0141
DBSCAN	0.99	-0.0063	0.0076	0.4493
Spectral	0.63	**0.0168**	**0.0129**	0.0134
AP	0.0	0.0031	0.0078	0.4615
BIRCH	**1.0**	-0.0061	0.0054	**0.5701**
Mean shift	**1.0**	-0.0063	0.0075	0.4535
OPTICS	0.99	0.0012	0.0019	0.4578
GMM	**1.0**	-0.0062	0.0054	0.0141
DEC	**1.0**	-0.0061	0.0055	0.0141

Table 9. Results for hypergraph-based embedding features of Bitcoin transactions.

Approach	SC	ARI	NMI	F1-score
k-means	**0.63**	0.0169	0.0129	0.3695
DBSCAN	-0.01	0.0139	0.0368	0.2775
Spectral	**0.63**	0.0168	0.0129	0.3694
AP	0.46	0.0608	0.0303	0.6047
BIRCH	**0.63**	0.0209	0.0141	0.5119
Mean shift	0.51	**0.0724**	0.0397	**0.6133**
OPTICS	-0.47	0.0678	**0.0654**	0.5782
GMM	-0.16	0.0004	0.00001	0.5640
DEC	0.45	0.0438	0.0184	0.4826

Clustering results obtained for graph embedding features of the Bitcoin transactions are listed in Tables 8, and 9. The results reveal that the clustering approaches k-means, spectral, and BIRCH obtained a Silhouette score of 1.0 for transaction graph-based embedding features and 0.63 for hypergraph-based embedding features. In addition, Mean shift, GMM, and DEC also received an SC score of 1.0 for transaction graph-based embedding features. The external validation scores of the embedding features were not promising, which reflects that the cluster identification does not correlate with the actual labels of the transactions. The generation of embedding features relies on establishing connections between the target node and its neighbours, which inherently empowers centrality, connectivity, and distribution-based clustering approaches to effectively discriminate between suspicious and normal transactions.

A Silhouette score of 1.0 associated with transaction graph-based embedding features signifies a robust separation between clusters. The ARI value of 0.7225 for structural features demonstrates a noteworthy correlation with the actual labels.

Table 10. Results for structural features of Ethereum Ponzi smart contract transactions.

Approach	SC	ARI	NMI	F1-score
k-means	0.47	0.0610	0.0383	0.7799
DBSCAN	0.06	0.0159	0.0025	**0.9154**
Spectral	**0.76**	0.0278	0.0026	0.3749
AP	0.59	0.0045	0.0253	0.0078
BIRCH	0.47	**0.0629**	0.0385	0.4264
OPTICS	0.38	-0.0001	0.0204	0.0332
Mean shift	0.47	0.0127	0.0284	0.4153
GMM	0.67	0.0508	0.0118	0.4428
DEC	0.46	0.03752	**0.0684**	0.0548

Table 11. Results for network property-based features of Ethereum Ponzi smart contract transactions.

Approach	SC	ARI	NMI	F1-score
k-means	0.5	0.0108	0.0084	0.5532
DBSCAN	0.33	-0.0085	0.0084	0.5377
Spectral	**0.71**	-0.0143	0.0035	0.5196
AP	0.62	0.0014	0.0072	0.0335
BIRCH	0.51	0.0032	0.0084	**0.6001**
OPTICS	0.25	-0.0132	**0.0088**	0.0134
Mean shift	0.6	-0.0002	0.0075	0.2646
GMM	0.57	**0.0109**	0.0007	**0.6001**
DEC	0.5	0.0094	0.0087	0.0623

Table 12. Results for hypergraph-based embedding features of Ethereum Ponzi smart contract transactions.

Approach	SC	ARI	NMI	F1-score
k-means	**0.63**	0.0025	0.0084	0.6042
DBSCAN	0.46	-0.0207	0.0080	**0.6533**
Spectral	0.59	-0.0095	0.0025	0.6270
AP	0.48	**0.0153**	0.0084	0.5472
BIRCH	**0.63**	-0.0014	0.0065	0.6287
OPTICS	0.16	-0.0252	0.0096	0.0167
Mean shift	**0.63**	0.0011	**0.0095**	0.2627
GMM	0.54	-0.0024	0.00029	0.0878
DEC	**0.63**	0.00923	0.0040	0.0592

Similarly, for Ethereum Ponzi smart contract transactions data, Table 10 summarises the results obtained. For the structural features, the spectral clustering approach obtained the highest Silhouette score of 0.76. The same approach obtained the highest Silhouette score of 0.71 for the network property-based features as shown in Table 11. The engineering of structural and network property-based features in the Ethereum network is rooted in establishing connections between wallets and their associated transactions. This inherent nature allows spectral clustering to differentiate between normal and suspicious wallets effectively

The clustering approaches k-means, BIRCH, Mean shift, and DEC obtained a 0.63 SC for hypergraph-based embedding features as detailed in Table 12. Based on the results, centroid, connectivity, and density-based clustering methods exhibited superior performance. The external evaluation values for all categories of features of the smart contract demonstrate a minimal agreement with actual labels.

The structural features of Ethereum Ponzi contracts achieved a desirable Silhouette score of 0.76, indicating a substantial separation between clusters; nonetheless, the near-zero external evaluation score implies a restricted correlation with the true labels.

The second phase of the analysis explored the importance of features using an explainable AI (XAI) approach. This phase used the Silhouette score to select the best clustering approach. The AP approach was chosen for the Bitcoin transaction structural and network property-based features. The k-means, spectral clustering, AP, BIRCH, GMM, and Mean shift approaches were selected for graph embedding features. For the Ethereum transactions, the spectral clustering approach was chosen for the structural and network property-based features. The k-means, BIRCH, and Mean shift clustering approaches were selected for the graph embedding features.

In the XAI analysis, we used the clusters found through specific clustering methods for each feature category. Using these clusters the average ShAP values are calculated to identify the top five influential features. The final phase of the analysis considered the above cluster evaluation results to (1) identify the best clustering approach for Bitcoin and Ethereum network analysis, (2) influential features for cluster identification, and (3) domain knowledge-based justification for the identified clusters. The results are discussed in the following section.

6 ANALYSIS AND DISCUSSION

This section critically analyses and discusses the findings reported in section 5 in view of cluster evaluation scores, explainable AI (XAI), and probability distribution of the features (domain knowledge).

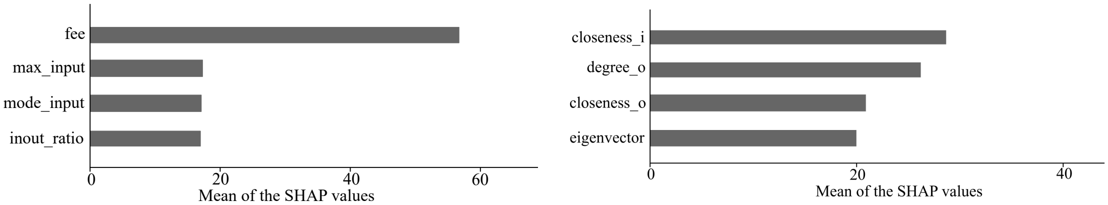
6.1 Analysis on Bitcoin transaction data clusters

The Silhouette scores for clusters formed by the clustering approaches for the structural and network property-based features are listed in Tables 6 and 7. These results reveal that the AP clustering approach efficiently separated Bitcoin transaction data into distinguishable clusters. For the identified clusters, the external evaluation scores ARI and NMI for structural and network property-based features confirm that the identified clusters are highly associated with actual labels. The other centroid-based clustering approaches were insufficient to distinguish the Bitcoin transactions.

Feature importance was analysed by computing the ShAP values on the identified clusters. Fig. 2a to 5b present the mean ShAP values of the top five structural and network property-based features. In all plots, the x-axis represents the mean of the ShAP values, and the y-axis lists the top five features arranged in descending order of importance.

The ransomware settlement in Bitcoin networks exhibits a pattern where funds are received from mixing services wallets, transferred to multiple intermediate mixing services, and ultimately spent on the ransomware wallet. This behaviour indicates that the features related to receiving and spending transactions can be influential in detecting suspicious activities. Notably, the ShAP values further determined the statistical measures of input and output features and transaction fees as influential structural features of the identified clusters, as depicted in Fig. 2a.

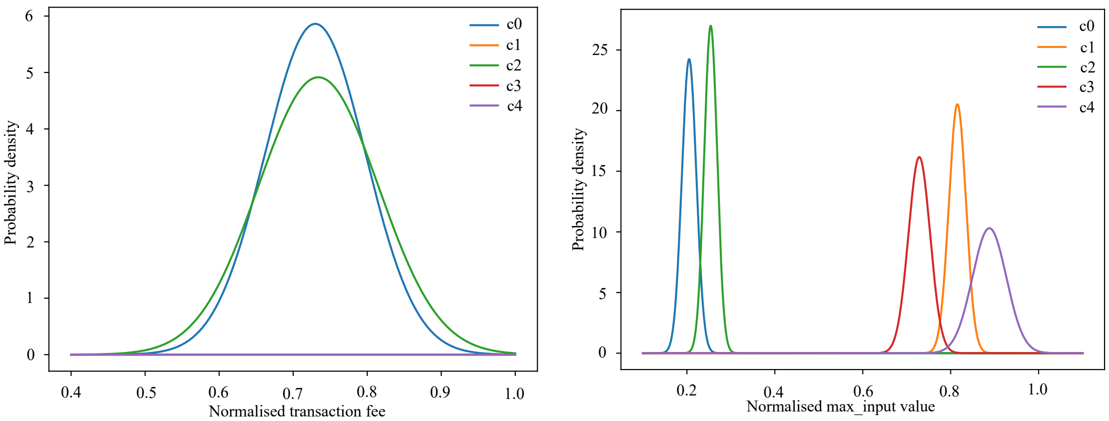
The ShAP values presented in Fig. 2b demonstrate the significance of degree, eigenvector, and closeness centrality measures in the identification of clusters using the features based on network properties. These centrality measures assess the influence of the nodes in proximity to other nodes. To illustrate the underlying significance of the identified clusters based on the structural and network properties of Bitcoin and smart contract transactions, this study investigates the distribution of the top two key features (Fig. 2 and Fig. 5) across the identified clusters. For the Bitcoin transactions, Figures 3a and 3b show the distribution of the transaction fee and max_input features within the clusters identified using the AP approach. Here, c0, c1, c2, c3, and c4 represent the identified clusters. As shown in Fig. 3a, three clusters are visible for the normalised transaction fee values within the 0.5 to 1.0 range. Similarly, five clusters are observed for the normalised values of the max_input feature within the 0.1 to 1.0 range. Figures 4a and 4b show the distribution of the network properties-based closness_i and degree_o features within the clusters identified using the AP approach. As presented in Fig. 4a, five clusters are visible for the normalised closness_i values within the range of 0 to 1.4. Similarly,



(a) Influential structural features.

(b) Influential network property-based features.

Fig. 2. Shapley values for the features of Bitcoin ransomware transactions based on affinity propagation outcomes



(a) Transaction (fee) feature of the Bitcoin transaction.

(b) The (max_input) feature of the Bitcoin transaction.

Fig. 3. Probability distribution of the transaction fee and max_input features in the **Affinity Propagation** based clusters

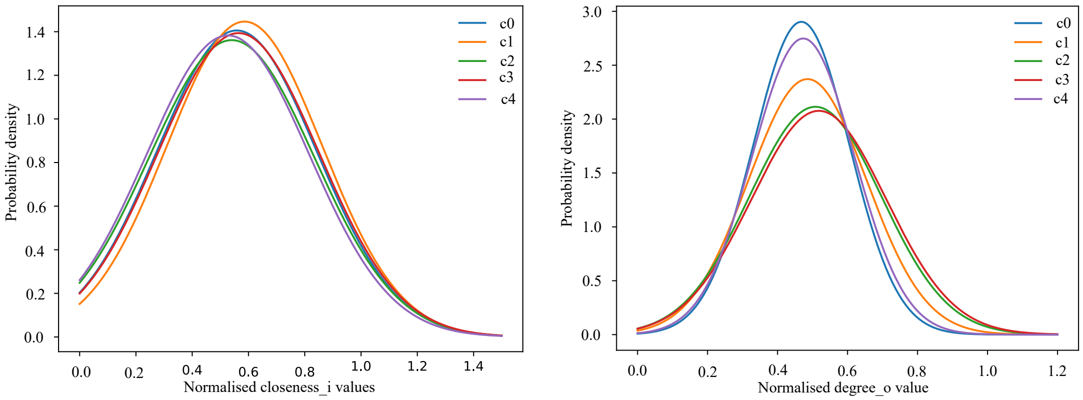
Table 13. Comparison of clustering results obtained from transaction graph-based embedding features and the related literature. Here, the columns precision (pre), recall (rec), and F2 (f2) results are calculated using actual labels and identified cluster IDs.

Blockchain	**Embedding**	**pre**	**rec**	**f2**
Bitcoin	**Proposed features**	0.600	0.977	0.868
	BTCOut [46]	0.589	0.264	0.473

five clusters are observed for the normalised values of the degree_o feature within the 0 to 1.1 range, as shown in Fig. 4b. Therefore, these visualisations help interpret the underlying characteristics of the identified clusters, providing insights into their actual meaning.

6.2 Analysis on Ethereum smart contract transactions data clusters

The Ponzi smart contracts exhibit a behaviour where funds were received from new investors and subsequently spent on older investors. This pattern emphasises the importance of the number of spending and receiving transactions and their corresponding values in identifying distinguished clusters within smart contracts. The higher Silhouette score for the clusters identified on Ethereum Ponzi contract data, stated in Tables 10, 11, and 12, reveals that the spectral,



(a) The (closness_i) feature of the Bitcoin transaction.

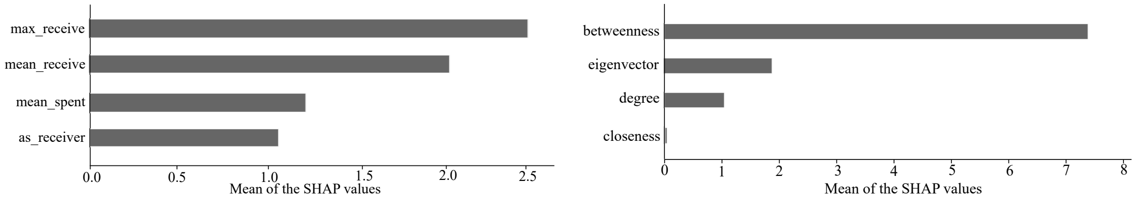
(b) The (degree_o) feature of the Bitcoin transaction.

Fig. 4. Probability distribution of the closness_i and degree_o features in the **Affinity Propagation** based clusters

BIRCH, Mean shift and DEC algorithms have identified distinguishable clusters. Anyhow, the cluster identification using mapping labels and calculating the ARI, NMI and F1-score revealed that they are less associated with actual smart contracts labels. The comparison between identified clusters and the actual labels reveals that in the clusters based on structural features using the spectral clustering approach, 39% of Ponzi contracts were clustered with non-Ponzi contracts. For network property-based features using the spectral clustering approach, 56% of Ponzi contracts were clustered with non-Ponzi contracts. For the embedding features using BIRCH, about 70% of the Ponzi contracts were clustered with the non-Ponzi contracts. The clusters identified through structural features of smart contracts exhibit a significant correlation with the actual labels, as indicated by the false positive percentage. In contrast, the network property-based and graph embedding features related to behaviour, pose difficulties in achieving similar correlation. It is notable, the dataset selected for smart contract clustering was experimented with for the first time. The actual labelling may themselves be ambiguous or subjective for the metadata information of the smart contract transaction, and this could potentially be one of the primary reasons for low ARI and NMI scores observed in Tables 10, 11, and 12.

Fig.5a provides the importance of features for clusters identified using statistical measures-based features. Similarly, Fig.5b presents the importance of features for clusters identified using the network property-based features. The probability density of these influential features is utilised to identify the meaning of the smart contract transaction-based clusters. Figures 6a and 6b show the distribution of the max_receive and mean_receive features within the clusters identified using the Spectral clustering approach. Here, c0 and c1 represent the identified clusters. As shown in Figures 6a and 6b, two clusters are visible for the normalised max_receive and mean_receive values within the 0 to 1.6 range. Similarly, Figures 7a and 7b also present two clusters for the normalised betweenness and eigenvector values within the range of 0 to 1.6.

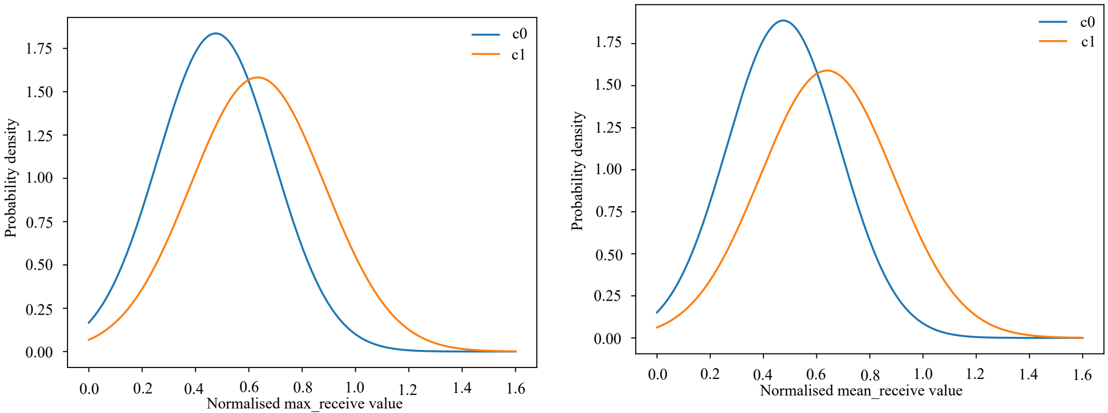
All clustering methods worked effectively for embedding features from the Bitcoin and Ethereum networks, while centroid-based methods like k-means, spectral clustering, and AP performed particularly well for structural features. The dataset used in this study is comparatively small. The transactions in the dataset are not related to one another and hence represent sparse points in the Bitcoin or Ethereum network. This sparsity prevents the embedding learning model from representing the transaction features properly. Consequently, the resulting embedding features provided less information when it comes to capturing complex patterns and relationships among both normal and malicious



(a) Influential structural features.

(b) Influential network property-based features.

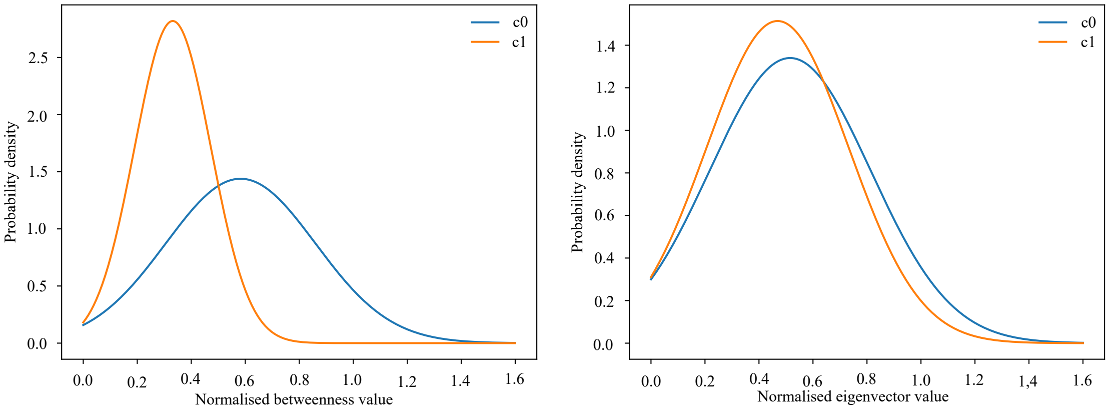
Fig. 5. Shapley values for the features of Ethereum Ponzi contract transactions based on spectral clustering outcomes



(a) The (max_receive) feature of the smart contract transaction.

(b) The **(**mean_receive) feature of the smart contract transaction.

Fig. 6. Probability distribution of the max_receive and mean_receive features in the **Spectral** based clusters



(a) The (betweenness) feature of the smart contract transaction.

(b) The (eigenvector) feature of the smart contract transaction.

Fig. 7. Probability distribution of the betweenness and eigenvector features in the **Spectral** based clusters

nodes in relation to other nodes. These less informative features pose a challenge for the AP clustering approach or any other approach in its task of distinguishing transactions. As a result, the AP clustering approach may have achieved lower internal and external evaluation scores, as listed in Tables 8, 9, and 12.

Overall, the results presented in Section 5 demonstrate the effectiveness of centroid-based clustering methods, k-means, spectral clustering, and Affinity Propagation (AP) in handling the structural features of Bitcoin transaction data. The analysis presented in Table 13 highlights the significant advancement achieved by this study for transaction graph-based embedding features. This study generates embedding features by employing structural and behavioural information of the Bitcoin transaction network described in section 4.2. This improves the quality of the embedding vector and enhances the precision, recall, and F2 scores of the identified clusters (for the GMM clustering approach) reported in the research [46]. Here f2 is an evaluation measure proposed in [46], also for comparison purposes the results are chopped to three digits. The row '**Proposed features**' indicates the way this research generates transaction graphs embedding features, and 'BTCOut' is the embedding approach reported in the literature [46]. The explainable clustering-based findings reveal that statistical measures of spending and receiving transactions' features are highly influential in the identification of two distinguished clusters.

6.3 Discussion

The primary limitation of this study is that the performance of clustering is evaluated using a small, static blockchain dataset. As a result, the results may not fully reflect the challenges associated with dynamic, temporal, and large-scale blockchain environments. Additionally, the current embeddings are generated for small-scale Bitcoin or Ethereum networks with 64 dimensions, which may not capture the full complexity of larger, more diverse blockchain ecosystems. The validation data in this research were labelled based on predefined conditions, which may not accurately capture true anomalous behaviours in blockchain transactions. Obtaining sufficient ground-truth data presents a significant challenge when implementing robust clustering algorithms and performing accurate validation. Despite these limitations, this study provides valuable insights into the initial performance of clustering algorithms on blockchain data, laying a strong foundation for future research in more complex and dynamic blockchain environments.

The analysis of the contribution of structural and network properties-based features to cluster identification, discussed in subsections 6.1 and 6.2, highlights that by monitoring key features in Bitcoin and smart contract transactions, institutions can detect suspicious behaviour early to prevent attacks. Additionally, this will strengthen smart contract security by mitigating risks and empowering regulators to create more targeted risk assessments and compliance strategies, ultimately reducing illicit activities and improving transparency.

> The centroid-based and connectivity-based clustering approaches demonstrate strong suitability for effectively distinguishing normal and anomalous behaviours.

7 CONCLUSION

This study provides valuable insights for researchers and practitioners seeking to develop unsupervised learning-based methods to analyse the suspicious behaviour of the actors in blockchain networks. The analysis of the various clustering approaches identified that connectivity-based and centroid-based clustering approaches are more suitable for blockchain binary class datasets. The findings using explainable AI (XAI) reveal that the statistical measures of spending and receiving transactions and the number of transactions connected with the nodes are the most valuable features for cluster identification. The embedding features of the blockchain transaction identified well-separated clusters but exhibited limited correlation with actual labels. Future research should focus on implementing the connectivity-based or centroid-based unsupervised learning model to identify malicious behaviour in real-time blockchain data. Another potential direction for future research is exploring alternative embedding techniques and refining clustering algorithms to enhance the performance of embedding-based malicious behaviour detection.

REFERENCES

[1] Cuneyt Gurcan Akcora, Yitao Li, Yulia R Gel, and Murat Kantarcioglu. 2019. BitcoinHeist: Topological Data Analysis for Ransomware Detection on the Bitcoin Blockchain. *arXiv preprint [Web Link]* (2019).

[2] Hosein Alizadeh, Behrouz Minaei-Bidgoli, and Hamid Parvin. 2014. Cluster ensemble selection based on a new cluster stability measure. *Intelligent Data Analysis* 18, 3 (2014), 389–408.

[3] Mihael Ankerst, Markus M Breunig, Hans-Peter Kriegel, and Jörg Sander. 1999. OPTICS: Ordering points to identify the clustering structure. *ACM Sigmod record* 28, 2 (1999), 49–60.

[4] Ilham Firman Ashari, Eko Dwi Nugroho, Randi Baraku, Ilham Novri Yanda, Ridho Liwardana, et al. 2023. Analysis of Elbow, Silhouette, Davies-Bouldin, Calinski-Harabasz, and Rand-Index Evaluation on K-Means Algorithm for Classifying Flood-Affected Areas in Jakarta. *Journal of Applied Informatics and Computing* 7, 1 (2023), 95–103.

[5] Imran Bashir. 2017. *Mastering blockchain.* Packt Publishing Ltd.

[6] Purnima Bholowalia and Arvind Kumar. 2014. EBK-means: A clustering technique based on elbow method and k-means in WSN. *International Journal of Computer Applications* 105, 9 (2014).

[7] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. 2000. LOF: identifying density-based local outliers. In *COMAD*. 93–104.

[8] Lu Cao and Hong Shen. 2022. CSS: Handling imbalanced data by improved clustering with stratified sampling. *Concurrency and Computation: Practice and Experience* 34, 2 (2022), e6071.

[9] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. 2018. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*. 132–149.

[10] Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2009. Anomaly detection: A survey. *ACM computing surveys (CSUR)* 41, 3 (2009), 1–58.

[11] Dorin Comaniciu and Peter Meer. 2002. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence* 24, 5 (2002), 603–619.

[12] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. Density-based spatial clustering of applications with noise. In *Int. Conf. knowledge discovery and data mining*, Vol. 240.

[13] Sean Foley, Jonathan R Karlsen, and Tālis J Putniņš. 2019. Sex, drugs, and bitcoin: How much illegal activity is financed through cryptocurrencies? *The Review of Financial Studies* 32, 5 (2019), 1798–1853.

[14] Huaqun Guo and Xingjie Yu. 2022. A survey on blockchain technology and its security. *Blockchain: Research and Applications* 3, 2 (2022), 100067. https://doi.org/10.1016/j.bcra.2022.100067

[15] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems* 30 (2017).

[16] Butian Huang, Zhenguang Liu, Jianhai Chen, Anan Liu, Qi Liu, and Qinming He. 2017. Behavior pattern clustering in blockchain networks. *Multimedia Tools and Applications* 76 (2017), 20099–20110.

[17] Marco Alberto Javarone and Craig Steven Wright. 2018. From Bitcoin to Bitcoin Cash: a network analysis. In *Proceedings of the 1st Workshop on Cryptocurrencies and Blockchains for Distributed Systems* (Munich, Germany) *(CryBlock'18)*. Association for Computing Machinery, New York, NY, USA, 77–81. https://doi.org/10.1145/3211933.3211947

[18] Samantha Jeyakumar, Zhé Hóu, Andrew Charles Eugene Yugarajah, MARIMUTHU PALANISWAMI, and Vallipuram Muthukkumarasamy. 2023. Visualizing Blockchain Transaction Behavioural Pattern: A Graph-based Approach. (3 2023). https://doi.org/10.36227/techrxiv.22329889.v1

[19] Samantha Tharani Jeyakumar, Andrew Charles Eugene Yugarajah, Zhé Hóu, and Vallipuram Muthukkumarasamy. 2024. Detecting Malicious Blockchain Transactions Using Graph Neural Networks. In *Distributed Ledger Technology*, Naipeng Dong, Babu Pillai, Guangdong Bai, and Mark Utting (Eds.). Springer Nature Singapore, Singapore, 55–71.

[20] Xin Jin and Jiawei Han. 2010. *K-Means Clustering.* Springer US, Boston, MA, 563–564. https://doi.org/10.1007/978-0-387-30164-8_425

[21] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. 2008. Isolation Forest. In *2008 Eighth IEEE International Conference on Data Mining*. 413–422. https://doi.org/10.1109/ICDM.2008.17

[22] Scott M. Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Long Beach, California, USA) *(NIPS'17)*. Curran Associates Inc., Red Hook, NY, USA, 4768–4777.

[23] Kevin Martin, Mohamed Rahouti, Moussa Ayyash, and Izzat Alsmadi. 2022. Anomaly detection in blockchain using network representation and machine learning. *Security and Privacy* 5, 2 (2022), e192.

[24] Aaron F. McDaid, Derek Greene, and Neil Hurley. 2013. Normalized Mutual Information to evaluate overlapping community finding algorithms. arXiv:1110.2515 [physics.soc-ph]

[25] Patrick Monamo, Vukosi Marivate, and Bheki Twala. 2016. Unsupervised learning for robust Bitcoin fraud detection. In *ISSA*. IEEE, 129–134.

[26] Pranav Nerurkar, Dhiren Patel, Yann Busnel, Romaric Ludinard, Saru Kumari, and Muhammad Khurram Khan. 2021. Dissecting bitcoin blockchain: Empirical analysis of bitcoin network (2009–2020). *Journal of Network and Computer Applications* 177 (2021), 102940. https://doi.org/10.1016/j.jnca.2020.102940

[27] Andrew Ng, Michael Jordan, and Yair Weiss. 2001. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems* 14 (2001).

[28] Kai Peng, Lixin Zheng, Xiaolong Xu, and Tao Lin. 2018. *Balanced Iterative Reducing and Clustering Using Hierarchies with Principal Component Analysis (PBirch) for Intrusion Detection over Big Data in Mobile Cloud Environment: 11th International Conference and Satellite Workshops, SpaCCS 2018, Melbourne, NSW, Australia, December 11-13, 2018, Proceedings.* 166–177. https://doi.org/10.1007/978-3-030-05345-1_14

[29] Thai Pham and Steven Lee. 2016. Anomaly Detection in the Bitcoin System - A Network Perspective. (Nov 2016).

[30] Blaž Podgorelec, Muhamed Turkanović, and Sašo Karakatič. 2020. A machine learning-based method for automated blockchain transaction signing including personalized anomaly detection. *Sensors* 20, 1 (2020), 147.

[31] Rizhen Qin, Lihua Zhao, Da Li, Ke Yang, Jiaxing Xuan, and Hejian Wang. 2021. Research on Design and Application of Power Dispatch Based on Blockchain. In *Proceedings of the 2021 3rd International Conference on Blockchain Technology* (Shanghai, China) *(ICBCT '21)*. Association for Computing Machinery, New York, NY, USA, 155–162. https://doi.org/10.1145/3460537.3460564

[32] Dorit Ron and Adi Shamir. 2013. Quantitative Analysis of the Full Bitcoin Transaction Graph. In *Financial Cryptography and Data Security*, Ahmad-Reza Sadeghi (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 6–24.

[33] Jorge M. Santos and Mark Embrechts. 2009. On the Use of the Adjusted Rand Index as a Metric for Evaluating Supervised Classification. In *Artificial Neural Networks – ICANN 2009*, Cesare Alippi, Marios Polycarpou, Christos Panayiotou, and Georgios Ellinas (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 175–184.

[34] Ketan Rajshekhar Shahapure and Charles Nicholas. 2020. Cluster Quality Analysis Using Silhouette Score. In *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*. 747–748. https://doi.org/10.1109/DSAA49011.2020.00096

[35] Congming Shi, Bingtao Wei, Shoulin Wei, Wen Wang, Hai Liu, and Jialei Liu. 2021. A quantitative discriminant method of elbow point for the optimal number of clusters in clustering algorithm. *EURASIP Journal on Wireless Communications and Networking* 2021, 1 (2021), 1–16.

[36] Priyanshi Singh, Deepika Agrawal, and Sudhakar Pandey. 2023. Anomaly detection and analysis in blockchain systems. (2023).

[37] Jeyakumar Samantha Tharani, Eugene Yougarajah Andrew Charles, Zhé Hóu, Marimuthu Palaniswami, and Vallipuram Muthukkumarasamy. 2021. Graph based visualisation techniques for analysis of blockchain transactions. In *2021 IEEE 46th Conference on Local Computer Networks (LCN)*. IEEE, 427–430.

[38] Jeyakumar Samantha Tharani, Eugene Yougarajah Andrew Charles, Zhé Hóu, Punit Rathore, Marimuthu Palaniswami, and Vallipuram Muthukkumarasamy. 2024. Unified Feature Engineering for Detection of Malicious Entities in Blockchain Networks. *IEEE Transactions on Information Forensics & Security* (2024). Accepted for publication.

[39] A. Wahrstatter, J. Gomes, S. Khan, and D. Svetinovic. 2023. Improving Cryptocurrency Crime Detection: CoinJoin Community Detection Approach. *IEEE Transactions on Dependable and Secure Computing* 20, 06 (nov 2023), 4946–4956. https://doi.org/10.1109/TDSC.2023.3238412

[40] Shuai Wang, Liwei Ouyang, Yong Yuan, Xiaochun Ni, Xuan Han, and Fei-Yue Wang. 2019. Blockchain-Enabled Smart Contracts: Architecture, Applications, and Future Trends. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 49, 11 (2019), 2266–2277. https://doi.org/10.1109/TSMC.2019.2895123

[41] Jiajing Wu, Jieli Liu, Weili Chen, Huawei Huang, Zibin Zheng, and Yan Zhang. 2021. Detecting mixing services via mining bitcoin transaction network with hybrid motifs. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 52, 4 (2021), 2237–2249.

[42] Jiajing Wu, Qi Yuan, Dan Lin, Wei You, Weili Chen, Chuan Chen, and Zibin Zheng. 2020. Who are the phishers? phishing scam detection on ethereum via network embedding. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* (2020).

[43] Junyuan Xie, Ross Girshick, and Ali Farhadi. 2015. Unsupervised Deep Embedding for Clustering Analysis. (11 2015).

[44] Yong Yuan and Fei-Yue Wang. 2018. Blockchain and Cryptocurrencies: Model, Techniques, and Applications. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 48, 9 (2018), 1421–1428. https://doi.org/10.1109/TSMC.2018.2854904

[45] Huan Zhang and Kun Song. 2011. Research and experiment on Affinity Propagation clustering algorithm. In *2011 Second International Conference on Mechanic Automation and Control Engineering*. 5996–5999. https://doi.org/10.1109/MACE.2011.5988401

[46] Rui Zhang, Guifa Zhang, Lan Liu, Chen Wang, and Shaohua Wan. 2020. Anomaly detection in bitcoin information networks with multi-constrained meta path. *Journal of Systems Architecture* 110 (2020), 101829.

[47] Yi Zhang, Miaomiao Li, Siwei Wang, Sisi Dai, Lei Luo, En Zhu, Huiying Xu, Xinzhong Zhu, Chaoyun Yao, and Haoran Zhou. 2021. Gaussian mixture model clustering with incomplete data. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 17, 1s (2021), 1–14.