Unified Feature Engineering for Detection of Malicious Entities in Blockchain Networks

Jeyakumar Samantha Tharani*, Zhé Hóu*, Eugene Yugarajah Andrew Charles†,
Punit Rathore§, Marimuthu Palaniswami‡, and Vallipuram Muthukkumarasamy*
*Griffith University †University of Jaffna ‡University of Melbourne §Indian Institute of Science
jeyakumar.samanthatharani@griffithuni.edu.au

***Abstract*—Blockchain technology has been integrated into a wide range of applications in various sectors, such as finance, supply chain, health, and governance. However, the participation of a few actors with malicious intentions challenges law enforcement authorities, regulators and other users. These challenges revolve around dealing with an array of illegal activities such as asset trades in dark markets, receiving payments for cyber-attacks, and facilitating money laundering. Developing an efficient mechanism to identify malicious actors in blockchain networks is a pressing need to build confidence among the stakeholders and ensure regulatory adherence. The raw data of blockchain transactions do not readily reveal the dynamic behavioural changes and their interconnection between transactions and accounts. These behavioural patterns can be useful for identifying malicious actors. Machine Learning (ML)-based models for early warning and/or detection are considered one of the potential approaches. In ML, feature engineering plays a crucial role in enhancing the predictive performance of a model. This study proposes different categories of features and unified feature extraction approaches for raw Bitcoin and Ethereum transaction data and their interconnection information. As far as we are aware, there has been no study that considered a feature engineering approach for identifying malicious activities. The significance of the engineered features was validated against eight classifiers, including Random Forest (RF), XG-boost (XG), Silas, and neural network-based classifiers. The results showed that these features contribute to higher classification accuracy and higher Area Under the Receiver Operating Characteristic Curve (AUC) value for both Bitcoin and Ethereum transactions. This work also analysed the influence of engineered features in classification using the eXplainable Artificial Intelligence (XAI) technique SHapley Additive exPlanations (SHAP) values. The feature importance scores confirmed the significance of the proposed engineered features towards implementing classification models to identify, target and disrupt malicious activities in blockchain networks.**

***Index Terms*—blockchain, Explainable Artificial Intelligence, cryptocurrency, anomaly detection, graph embedding**

I. Introduction

Blockchain technology has many potential applications in different industries [1], [2] that gives decentralisation, provenance and immutability for trustless transaction activities. In permissionless blockchain networks, participants can be pseudo-identified through public keys. This pseudonymous nature facilitates users to hide their real identities and/ or protect their privacy when performing transactions. This advantage also comes with caveats, as an enabler for *malicious activities*, such as Bitcoin payments for ransomware attacks [3], black market trades and facilitating money laundering. Cybercriminals exploit anonymity property together with the borderless distributedness of the blockchain to easily cover up traces of illegal activities. Recently, the Cerber ransomware campaign [4] demanded ransomware payments in Bitcoin. In this event, tens of thousands of victims transferred their Bitcoins to a single wallet. From there, the Bitcoin payments were transferred into a large number of other accounts, facilitating a form of money laundering. The ability to analyse the behaviour of such transaction networks, in almost real-time, would pave an effective way to taking down malicious entities and identifying the actual source behind such activities.

The transactions in a blockchain network can send and receive cryptocurrencies or asset tokens and create or invoke a smart contract. Each of these transactions has its own properties and an interconnection (spend or receive) with other transactions. The interconnections between the transactions represent the behaviour of the entities (accounts or smart contracts) in the blockchain network. The properties of transactions and their behavioural information are the potential inputs for identifying malicious entities. In general, the identification of malicious entities involves exploiting domain knowledge, forming logical reasoning, feature engineering and, most of all, a time-consuming process to identify the best-performing features and machine learning models. Identifying malicious entities in blockchain networks also needs meaningful transformation of the transaction data into an appropriate domain that can easily support the needed analysis. Various studies analysed blockchain transactions based on path, connectivity, community, and node analysis [5]–[8]. These studies mainly utilised address-based features of blockchain transactions. The extraction of address-based features involves domain knowledge of the address holders and clustering heuristics [7], [9], [10]. The clustering heuristics are, however, task-specific and cannot achieve similar performance while employing different or unknown tasks. When these heuristics perform the analysis on large-scale industrial networks, they may also suffer from high computational costs and excessive memory requirements. Considering these limitations, this research proposed an automated unified feature engineering pipeline for extracting feature values from blockchain networks. Feature engineering involves obtaining features from raw transaction data, aggregating statistical measures of the extracted features, transforming transaction data into a graph, obtaining network properties (using node centrality measures), and generating embedding features using graphs. These features obtained

significantly high test accuracies for classifying malicious behaviour of Bitcoin transactions (88.0%) and Ethereum EOAs (97.86%). In addition, this work analysed the influence of a feature or set of features in malicious behaviour identification using the XAI technique SHAP values. The main contributions of this work are:

1) Identified novel structural and relational features of the Bitcoin transactions and Ethereum's Externally Owned Accounts (EOAs) in this work. The validation results based on the performance of standard machine learning techniques for detecting malicious transactions and EOAs confirmed that the proposed features achieve a higher F1-score compared to the features in existing datasets sourced from the same raw data.
2) Proposed a unified automated feature engineering pipeline to extract feature values from raw Bitcoin/ Ethereum transaction data and their interconnection information. To the best of our knowledge, this is the first work which automates blockchain data engineering in a systematic way that is applicable to multiple types of blockchain networks.
3) The results of the XAI-based analysis revealed the influential features which are highly correlated with the behaviour of blockchain transactions. This sheds light on the features which are significantly correlated with suspicious behaviour and are beneficial for training efficient classifiers.

The rest of the paper is structured as follows: Section II presents a critical review of related work. Section III describes an overview of the proposed unified feature engineering approach, and Section IV explains the process of data collection from blockchain networks. Section V details the feature extraction, graph modelling and algorithms involved in features engineering. Section VI presents results for engineered feature validation using the classification approaches. Section VII analyses the effectiveness of engineered features in classification using the SHAP feature importance measure. Section VIII discusses the research findings, limitations, and future directions. Finally, Section IX concludes the paper.

II. Related Work

This section summarises the literature on malicious activity detection in blockchain networks. Specifically, this section details the features used in the literature for the said purpose. Most of the research used address-based features such as structural (based on metadata information), centrality scores, and topological and geometric structures of the user or entity graphs to analyse the characteristics of transactions in a specific blockchain.

Akcora et al., [5] proposed a classification approach based on topological and geometric features of a Bitcoin transaction graph. Their proposed method used an obfuscation pattern of the Bitcoin transaction to predict ransomware and normal addresses. They published Bitcoin normal and ransomware-related address-based dataset BitcoinHeistRansomwareAddressDataset [11] with six features. These six address-based features were identified using domain knowledge of ransomware transactions and clustering heuristics. Moreover, their proposed heuristics are task-specific (based on splitting and merging) and do not achieve similar performance across different or unknown tasks.

Pranav et al., [7] proposed an ensemble of Decision Tree (DT) approaches to classify malicious actors in the Bitcoin network. Their proposed approach considered nine features based on the amount and time of transactions and addresses in the Bitcoin network. There is no result reported regarding the influence of the features and the type of malicious behaviour classified. The dataset used in their experiment is also not publicly available to compare the impact of the features.

Weber et al., [12] published an Elliptic dataset from a graph network of Bitcoin transactions. Their proposed graph network maps the Bitcoin transactions of real entities belonging to licit categories (exchanges, wallet providers, miners, licit services) and illicit categories (scams, malware, terrorist, organizations, ransomware, Ponzi schemes). In their constructed graph, nodes represent transactions and the edges represent the flow of Bitcoin (BTC) from one transaction to the next. In the Elliptic dataset, 2% are labeled illicit and 21% are labelled licit the remaining transactions are labelled as unknown. Each node in the graph is associated with 166 features and the first 94 features represent local information about the transaction such as time step, number of input/output, transaction fee, and output volume. The remaining 72 features are called aggregated features. Their proposed dataset achieved 97.1% precision, 67.5% recall, and 79.6% F1-score for the RF classifier. It is noteworthy that the class imbalance in the Elliptic dataset could be the cause of the low recall and precision, despite achieving a high precision. The major limitation of the Elliptic dataset is that the feature set details are anonymised; making it impossible to identify the features influencing the classification. Their work did not report any results regarding the influence of the transaction features in the classification.

Lorenz et al., [13] used the Active Learning (AL) approach to detect money laundering-related transactions in the Bitcoin network. They used the Elliptic [12] dataset for their experiment. The major limitation of the Elliptic dataset is that the feature set is anonymised; due to this reason, their experiment has not reported any results regarding the influence of the features in the AL-based money laundering-related nodes' classification.

Elmougy et al., [14], published the Elliptic++ dataset as an extension of the Elliptic dataset. It contains 822k Bitcoin wallet addresses with 56 features and 203k Bitcoin transactions with 183 features. The values of transaction and wallet features published in this dataset mainly focus on money laundering activities. Their transaction dataset achieved 97.5% precision, 71.9% recall, and 82.8% F1-score for the RF classifier. The limitations in the Elliptic dataset are applicable here as well.

Song et al., [15] published an HBTBD heterogeneous Bitcoin transaction dataset for predicting anti-money laundering in the Bitcoin network. This only considers licit and illicit transactions of the Elliptic dataset [12]. In addition to the features of the transactions at the Elliptic dataset, HBTBD obtained another set of features using the Bitcoin transactions and addresses metapaths. The metapath feature of the address is calculated based on the input and output relations between

the transaction and wallets. The features identified in this dataset achieved 92.2% precision, 61.7% recall, and 73.9% F1-score as the best results for the illicit nodes' classification using Random Forest+MAGNN. The MAGNN is a heterogeneous graph neural network. In their dataset, the real context information of the wallet address features is not available for validation.

Monamo et al., [16] detected fraudulent activity in Bitcoin transactions using an unlabelled dataset (URL not available for validation) created by the Laboratory for Computational Biology at the University of Illinois. The data used for their experiment contains fourteen Bitcoin transaction-based features, considering currency-based, network features-based, and the average neighbourhood-based aspects. Their results only evaluated the trimmed k-means clustering for unsupervised cybercrime detection in the Bitcoin network. However, their investigation did not include any comprehensive examination or disclosure of the influence and effects of the engineered features employed during the analysis.

Pham et al., [17] investigated the Bitcoin network analysis using three unsupervised learning-based network analysis approaches - power degree and densification laws, k-means clustering, and Local Outlier Factor (LOF) [18]. 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 reveals the anomalous behaviour for power degree distribution values, and the mean of incoming and outgoing transaction amount features support identifying anomalies using the LOF approach. There are no specific findings reported for the k-means clustering. Noticeably, the dataset used in their research work is not publicly available for comparison.

Podgorelec et al., [19] used features based on the minimum, maximum, mean and standard deviation of the number of transactions from Ethereum mainnet within a given time-frame. Their identified features were used for the anomalous behaviour detection of smart contract or externally owned accounts using the Isolation Forest unsupervised learning approach [20]. However, they have not reported any details regarding the impact of the identified features in anomaly detection. The dataset they used for the experiment is unlabelled, lacks detailed information about the features, and is publicly unavailable for comparison.

Zhang, Rui et al., [6] used the Bitcoin transaction dataset provided by the ELTE project [21] to evaluate their $BTCOut$ algorithm. The dataset contained unlabelled 11 million users (addresses) and 19 million transaction records. The features of the transaction used in their research were based on time, amount, and structural constraints. Their unsupervised-based anomaly detection used 18 features of the transaction and user graphs. However, they did not report any analysis regarding the contribution of their selected features for anomaly detection.

Scicchitano et al., [22] used historical logs of the Ethereum Classic network to detect the Decentralised Autonomous Organisation (DAO) attacks using an unsupervised learning-based encoder-decoder deep learning model. They released `Ethereum Classic Blockchain` dataset [23], which contains blocks, transactions, contracts, logs, token transfers, tokens, and traces. Their experiment only validated the data related to block using twelve features related to block size and related transactions. Based on the result, their engineered features are insufficient to detect the DAO attack.

Samantha et al., [24] used the structural features of the Bitcoin transactions namely inDegree, outDegree, totalInput, and totalOupt as node properties for hypergraph-based detection of malicious participants. They obtained improved performance for the node classification using the above features.

Wu, Jiajing et al., [25] proposed a network embedding algorithm `trans2Vec` to classify phishing transactions of EOAs in the Ethereum network. Their node embedding algorithm utilised the amount and time-based features of the EOAs to construct an embedding vector (64 dimensions). The embedding vectors constructed by their proposed approach were validated using the Logistic Regression (LR), Naive Bayes (NB), Isolation forest, and one-class Support Vector Machine (SVM) [26] and obtained a reasonable F1-score. Their findings reported that the transaction networks' structural, time and amount-based information influenced the detection of phishing.

In the literature, analysis of blockchain transactions mostly considered structural, time-based, centrality measures-based features of the wallets and topological and geometric-based features of user or entity graphs. Most of the datasets used in related studies noticeably lack public accessibility, which hinders validation. The usability of the existing Elliptic [12] dataset is limited since the number of anomalous transactions is relatively small compared to normal ones. This can make it challenging for machine learning models to learn and generalise to identify unknown anomalies. Another limitation is the lack of context for features. The absence of context makes it difficult to understand the reasons behind the labels of transactions. Both Elliptic and Xblock datasets [25] may not fully capture the temporal dynamics of financial transactions which limits the ability of models to detect anomalies over time. The other dataset BitcoinHeist [10] for normal and various ransomware wallet settlements comprises six graph-based features. Considerably, the features in BitcoinHeist were extracted based on the heuristics and the labelling of ransomware wallets was made on certain assumptions. These subjective and uninformative features considered in the literature underscore the need for systematic feature engineering for the analysis of blockchain networks. Recognising these gaps and challenges, this research aims to introduce a unified feature engineering to extract various categories of features of blockchain transactions. Further, the significance of these features is validated through the application of XAI techniques in identifying nodes involved in illicit activities.

III. Overview of the Proposed Methodology

This section outlines the methodology of the proposed unified feature engineering approach, including data collection, extracting features from blockchain transactions and validating them using various machine learning models and feature

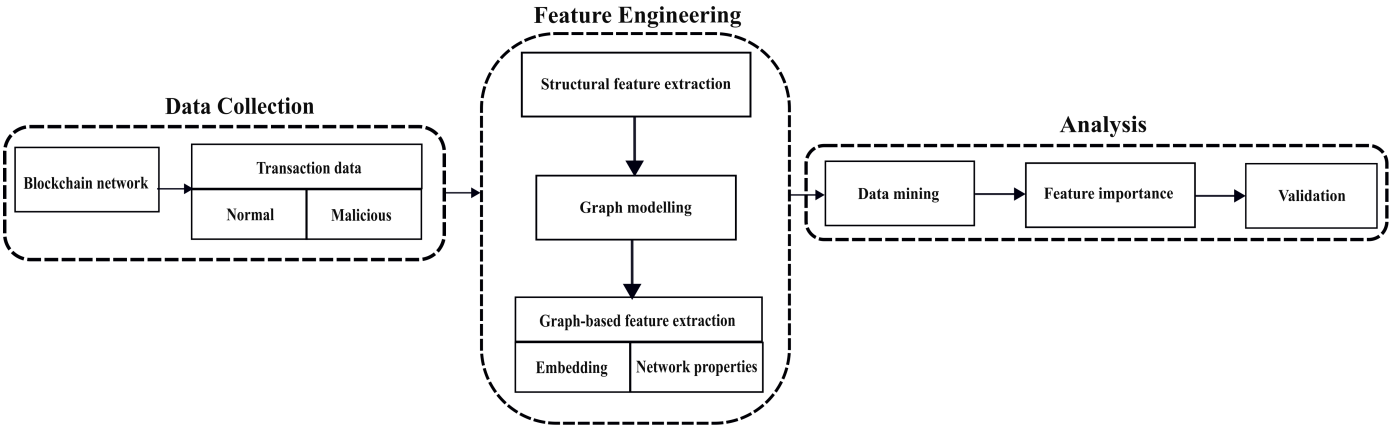


Fig. 1: Overview of the proposed methodology.

importance scores. As represented in Fig. 1, the proposed methodology contains three main phases: (i) data collection, (ii) feature engineering, and (iii) analysis. These phases clearly describe the main tasks involved in this research work and group the techniques and processes associated with them.

The data collection phase performed the fetching of raw transactions (JSONs) related to normal and malicious nodes from Bitcoin and Ethereum networks (transaction or EOA address). A detailed explanation of the processes within this phase is given in Section IV. The outcome of this phase is fed into the second phase dedicated to feature engineering.

In the feature engineering phase, transaction data obtained via the data collection are used to engineer three sets of features namely, structural, graph-based network properties, and embedding. For the first part of feature engineering, several raw transaction feature values are extracted and then used to generate aggregated feature values by applying different statistical measures. The combination of raw and aggregated features is referred to as structural features. A detailed explanation for structural feature extraction is given in Section V-A. For the second part of feature engineering, the transaction JSONs are transformed into various graphs on top of the graph database that involves various algorithms [27]. These graphs capture the behavioural features of transactions and EOAs for different types of blockchain networks (Unspent Transaction Output - UTXO or account-based). Graph modelling on top of the graph database eliminates the need for intermediate processes like data normalisation. The process involved in graph modelling for blockchain transactions is detailed in Section V-B.

The outcome of graph modelling is used to capture graph-based network property features and embedding features. Graph-based network properties and embedding features are a vector representation of the relationship between a node and its neighbours. The embedding with higher dimensions increases the computational cost but generally yields richer information in the embedding vector. This research chose a dimension of 20 for the embedding vector to balance the required computational resources and information to detect malicious activities. The detailed description of this phase is described in Sections V-C and V-D.

The final phase deals with the analysis of the engineered features from the previous phase. This phase evaluates the significance of engineered features via the results of node (transactions or EOAs) classification and feature importance analysis using the XAI technique, which identifies influential features related to malicious activities. The detailed processes involved in this phase are given in Sections VI and VII.

IV. Data collection

This section describes the collection of normal and malicious blockchain transaction data using public APIs. The APIs used in this research ([28] and [29]) have usage limitations, permitting only up to 250 transactions per wallet and a maximum of 100 requests per day. The APIs from various platforms provide transaction data in different formats; this will change the reference name of the key features in the generalised graph modelling algorithm. Additionally, the null or invalid values in the transactions provided via API affect feature aggregation and efficiency in graph-based modelling. Based on our knowledge, the design of blockchain transactions is often associated with pseudo-anonymity and the participants are identified by cryptographic addresses rather than personal information, but the linkage between these addresses and real-world identities may be established through various means [30] that may compromise privacy. Notably, it is not possible to infer any real-world identities related to transactions and the EOAs data used in this research work, ensuring privacy.

A. Datasets for Bitcoin network

Two Bitcoin datasets are created in this research. The first one consists of binary classes (normal and ransomware settlement-based) transactions and the other has multiple classes (theft, hack, dark market-based) of malicious transactions. The binary class dataset is created with normal and ransomware settlement-related Bitcoin transactions using the labelled ransomware addresses available in the BitcoinHeist [10] dataset. Recent 100 transactions corresponding with labelled addresses were captured using public API [28]. The assumptions made here are that all the transactions related to the normal addresses are non-suspicious and those related to ransomware settlements-related addresses are suspicious.

The multiclass malicious Bitcoin transactions dataset was created using raw transaction information of the hashes captured via public API [28]. The transactions were labelled based on the published IEEE-bitcoin malicious dataset [31]. Unlike the BitcoinHeist [10] dataset, which has only the addresses, the IEEE-bitcoin malicious dataset contains transaction hashes related to 1st FBI Silk Road seizures (542), 2nd FBI Silk Road seizures (609), Bitcoin thefts (55), and Bitcoin hack (47).

B. Dataset for Ethereum network

The dataset for normal and phishing settlements-related EOA addresses was prepared using the transaction information captured via public API [29]. The EOAs and their transactions were labelled based on the information available in the published first-order transaction data of the EOA addresses [25] which contain 1,660 phishing and 1,700 normal addresses with 9 features. The rest of the transaction information related to gas value and transaction fee was captured using public API [29].

V. Feature Engineering

This section describes the approach to engineering three categories of feature sets from the blockchain transaction data mentioned in Section IV. Subsection A describes the process of extracting structural features from Bitcoin and Ethereum data. Subsection B describes the algorithms related to graph modelling. Subsections C and D explain the approaches involved in graph-based feature extraction of network properties and graph embedding, respectively.

A. Feature extraction: Structural properties

The structural properties of a transaction may vary based on the type of the blockchain network. This section describes the process of extracting structural properties from the Bitcoin and Ethereum transaction data. It involves capturing raw features directly from the blockchain transaction and other aggregated features via statistical measures (maximum, minimum, mean, median, mode, and standard deviation).

1) Bitcoin transaction: This study focuses mainly on the transaction-based features of Bitcoin data. A total of 16 features were selected for the analysis. Among these, 12 features are newly created in this work. The features inDegree and outDegree, are reported in [32] and totalInput and totalOutput are reported in [24]. The details of the features are presented in Table I. It should, however, be noted that the *hash, lockTime, blockNumber, time, and isCoinbase* features are included to inform the features available in the raw transaction data, but are not used for classification.

2) Ethereum EOA: This study focuses mainly on the EOA-address-based features of Ethereum data. A total of 52 features were selected for the analysis. Among these, 50 features are newly introduced in this work, and the other two features, standard deviations of the received and spent amounts, are from the previous work reported in [33]. The equations (1), (2), (3), and (4) explain four main features (*asSender*, *asReceiver*, *totalSpent* and *totalReceived*), which contribute to the property of the EOA are given below:

TABLE I: Structural features for Bitcoin transaction.

Feature	Description
$hash$	unique identifier of the transaction
$lockTime$	limit of waiting time from the arrival of the transaction
$blockNumber$	unique number of the block which contains the transaction
$time$	confirmation time of the transaction
$isCoinbase$	feature indicating if the transaction is coinbase
$inDegree$	number of incoming transactions
$outDegree$	number of outgoing transactions
$totalInput$	total amount of Bitcoins received
$totalOutput$	total amount of Bitcoin sent
$(*) - input$	(min/ avg/ max/ med/ mod/ std) amount of Bitcoin received
$(*) - output$	(min/avg/max/med/mod/std) amount of Bitcoin spent

$$asSender = f(u_x), \text{ where } u_x \in S \tag{1}$$

$$asReceiver = f(u_x), \text{ where } u_x \in R \tag{2}$$

$$totalSpent(u_x) = \sum_{i=1}^{m} |t_{u_x v_i}|, \text{ where } v_i \in R \tag{3}$$

$$totalReceive(u_x) = \sum_{i=1}^{n} |t_{v_i u_x}|, \text{ where } v_i \in S \tag{4}$$

Where $f(u_x)$ represents the total number of times the node u_x participated as a sender or receiver, S is a list of senders, and R is a list of receivers. In equation (3) $|t_{u_x v_i}|$ is the amount spent by node u_x to node v_i. Whereas, in equation (4) $|t_{v_i u_x}|$ is the amount received by node u_x from node v_i. Where u_x and v_i are EOA addresses and m and n are the total number of spending and receiving transactions, respectively. The rest of the 48 features are created by aggregating the values of these four base features using statistical measures. Table II presents a detailed description of these structural features.

In Table I and Table II, the $(*)-$ represents the six statistical-based (maximum, minimum, mean, median, mode, and standard deviation) values of the given feature. The time complexity of structural feature extraction is expressed as $O(n)$, where n is the total amount of data in each category (transaction or EOA).

The structural features alone are not sufficient to characterise the interactions between the transactions. Capturing the intrinsic nature of these interactions needs appropriate representation. This research chooses graph modelling as an approach to achieve this need. The Algorithm 1 is designed to construct the graphs based on the type of blockchain network.

TABLE II: Structural features for Ethereum EOA.

Feature	Description
$asSender$	number of times a specific address as a sender
$asReceiver$	number of times a specific address as a receiver
$totalSpent$	total amount spent by a specific address
$totalReceive$	total amount received by a specific address
$(*) - spent$	(min/avg/max/med/mod/std) amount spent by a specific address
$(*) - receive$	(min/avg/max/med/mod/std) amount received by a specific address
$(*) - gasUsed\_s$	(min/avg/max/med/mod/std) gas used for sending
$(*) - gasUsed\_r$	(min/avg/max/med/mod/std) gas used for receiving
$(*) - gasPrice\_s$	(min/avg/max/med/mod/std) gas price used for sending
$(*) - gasPrice\_r$	(min/avg/max/med/mod/std) gas price used for receiving
$(*) - fee\_s$	(min/avg/max/med/mod/std) transaction fee for sending
$(*) - fee\_r$	(min/avg/max/med/mod/std) transaction fee for receiving

B. Graph modelling for blockchain transaction

In a blockchain network, the blocks are arranged as an ordered list, where each block is a collection of transactions. A transaction normally represents a record of transferring a digital asset between the sender and receiver. The sender and receiver can be transactions, wallets, EOAs or smart contracts. The nature of the asset depends on the type of blockchain. For example, in the Bitcoin network, the asset is a cryptocurrency in Bitcoin (BTC). Similarly, in the Ethereum network, the asset is a cryptocurrency in ether (ETH) or a token. In this research, the graph modelling considers interactions between transactions of the Bitcoin network and EOAs of the Ethereum network. The feature engineering from smart contract-based transaction data is similar to EOA [24], but this work only considered EOA. The graph modelling for smart contract-based transactions is also the same as the graph for EOA; only the type of the node will be different. The node properties for the EOA are applicable for smart contract addresses as well.

1) Transaction graph: A transaction graph is a homogeneous type of graph that applies to UTXO-based blockchain networks. A transaction graph $G_t = (V, E)$, where V represents the set of transactions and $E \subset V \times V$ represents the set of edges. Each transaction node has a set of node features $\mathcal{F} = \{inDegree, outDegree, totalInput, totalOutput\}$. The $e(u, v)$ represents the amount of bitcoin transferred between u and v.

2) Money flow transaction graph: The money flow transaction (MFT) graph is a homogeneous graph. It is applicable for both UTXO and account-based blockchain networks. In this research, the MFT graph $G_m = (V, E)$, where V represents the set of EOAs, and $E \subset V \times V$ is a set of edges. Each EOA node has a set of features, $\mathcal{F} = \{asSender, asReceiver, totalSpent, totalReceive\}$. The represents the amount of cryptocurrency transferred between u and v and the confirmation order of the transaction.

The lines from 5-22 and 23-35 in graph modelling Algorithm 1 incorporate both aspects described in subsections V-B1 and V-B2, respectively. It begins by reading the blockchain transactions and identifying the type of blockchain network (Bitcoin or Ethereum). Based on the type, it goes through the list of transactions individually and creates the nodes with their properties as stated either in subsections V-B1 or V-B2. The next step is creating edges between the nodes based on the transaction information. The edge information $e(u, v)$ varies based on the type of the blockchain network. The $txIndex$ variable defined in line 31 of Algorithm 1, is an index of the transaction which is calculated using the function $findIndex(tnxi_{hash})$. The function $findIndex$ returns an index for a given transaction hash from a sorted (ascending order) transaction list. Here sorting is carried out based on the confirmation time of transactions. Here, the $tnxList$ is the list of transactions. The time complexity of the graph modelling algorithm is primarily determined by the size of the transaction list, denoted as N. As described in Algorithm 1, when modelling Bitcoin transactions, the outer loop iterates N times, the first inner loop runs $(N - 1)$ times, and the second inner loop runs M times to establish connections between transactions, where M represents the number of inputs (UTXOs). In most cases, the value of M is insignificant compared to the value of N ($M \ll N$). As a result, the time complexity can be expressed as $O(N^2)$. In the case of Ethereum transactions, where there are no direct connections between the transactions, only the outer loop influences the time complexity, which is $O(N)$.

C. Feature extraction: Network properties

The network properties are considered powerful features for representing complex networks or graphs [34]. Graph modelling represents blockchain transactions as graphs. This work considers a node's centrality measures under the category of network properties only for the Ethereum network. The centrality measures are used to effectively capture the interactions of nodes in a graph quantitatively. The centrality measures of degree, betweenness, closeness, and eigenvector are calculated for the EOA to inform their influence or importance in the MFT. These are described below:

1) Degree centrality: The degree centrality represents the popularity of a node within a graph based on the total number of nodes connected to it. In this research, the degree centrality $C_d(u_x)$ in equation (5) represents the number of interactions of a particular EOA u_x with other EOAs $\{v_i\}$.

$$C_d(u_x) = \sum_{i=1}^{k} a_{u_x v_i}, \quad \text{where} \quad a_{u_x v_i} \in A \tag{5}$$

The matrix A represents the connection between node u_x and other nodes $\{v_i\}$ and $a_{u_x v_i}$ is an element in matrix A. The time complexity of $C_d(u_x)$ is $O(k+|E|)$, where k is the total number of nodes (EOA) and $|E|$ is the total number of edges in graph G.

Algorithm 1: Graph modelling for blockchain transactions.

Input: $tnxList$, $type$
Output: G

```
constructGraph(tnxList)
G ← {}
for i ∈ [1, size(tnxList)] do
    tnxi ← tnxList[i]
    hashi ← tnxi_hash
    if type == Bitcoin then
        for j ∈ [i + 1, size(tnxList)] do
            tnxj ← tnxList[j]
            inputsj ← tnxj_inputs
            hashj ← tnxj_hash
            for ins ∈ inputsj do
                sj_hash ← ins_hash
                if sj_hash == hashi then
                    s ← (sj_hash, inDegree_s, outDegree_s,
                      totalInput_s, totalOutput_s)
                    r ← (hashj, inDegree_r, outDegree_r,
                      totalInput_r, totalOutput_r)
                    e(s, r) ← ins_value
                    V ← (s, r)
                    G ← (V, e(s, r))
                end
            end
        end
    else
        address_s ← tnxi_from
        address_r ← tnxi_to
        s ← (address_s, asSender_s, asReceiver_s,
          totalReceive_s, totalSpent_s)
        r ← (address_r, asSender_r, asReceiver_r,
          totalReceive_r, totalSpent_r)
        value ← tnxi_value
        txIndex ← findIndex(tnxi_hash)
        e(s, r) ← (value, txIndex)
        V ← (s, r)
        G ← (V, e(s, r))
    end
end
return G
```

2) Betweenness centrality: The betweenness centrality $C_b(u_x)$ measure is calculated based on the number of times an EOA u_x acts as a bridge along the shortest path between two other nodes $\{v_i\}$ and $\{v_j\}$ which is described in equation(6).

$$C_b(u_x) = \sum_{\substack{(v_i,u_x,v_j)\\ i\neq j}} \frac{f(\psi_{v_i\hat{u_x}v_j})}{f(\psi_{v_iv_j})} \tag{6}$$

Where, $\psi_{v_iv_j}$ is the shortest path between nodes v_i and v_j. The $f(\psi_{v_iv_j})$ is the number of shortest paths between v_i and v_j, and $f(\psi_{v_i\hat{u_x}v_j})$ is the number of the shortest paths between v_i and v_j that passes through node u_x, where, (v_i, u_x, v_j) is an ordered triple. The time complexity of $C_b(u_x)$ is $O(k*(k+|E|))$. The graph of the blockchain network exhibits a substantial volume for both k and $|E|$, the expression $k*(k+|E|)$ suggests an increase in the complexity of the betweenness centrality calculation. Node (EOA) with higher betweenness centrality has more influence over the transactions between other nodes.

3) Closeness centrality: The closeness centrality in equation (7) measures how close a node is to other nodes in a graph.

$$C_c(u) = \frac{1}{\sum_{v\neq u} e(u,v)} \tag{7}$$

where $e(u, v)$ is considered as the distance (value transferred) between node u and v in graph G. The time complexity of $C_c(u)$ is $O(k*(|E|+k*log(k)))$, where $k*log(k)$ is the complexity for calculating the shortest path from a source node to other. The node with the highest closeness centrality score has the shortest distances to all other nodes.

4) Eigenvector centrality: The eigenvector centrality $C_e(u)$ of a node in equation (8) is defined as the weighted sum of the degree centralities of all vertices that are connected to that particular node by an edge.

$$C_e(u_x) = \frac{1}{\lambda}\sum_{i=1}^{k} a_{u_xv_i} C_d(v_i), \text{ where } a_{u_xv_i} \in A \tag{8}$$

The matrix A represents the connection between EOA u_x and other EOAs $\{v_i\}$, λ is the largest eigenvalue of A and $a_{u_xv_i}$ is an element of matrix A. Where, $C_d(v_i) = (C(1), C(2), ..., C(k))^T$ is an eigenvector corresponding to the value λ. The $C_d(v_i)$ is calculated based on the incoming neighbours of node v_i. The time complexity for $C_e(u_x)$ is $O(j*(k^2))$, where j is the total number of iterations and k is the total number of incoming nodes of node v_i; in this work, the value of j is 20. A node with a higher eigenvector centrality score implies that the node is involved in many transactions.

D. Feature extraction: Graph embedding

Graph embedding is an effective way to represent the structure of a graph as a set of vectors. It can capture the topology, vertex-to-vertex relationship, and other relevant information about graphs. Equation (9) describes calculation of embedding, $\mathcal{H}(u)$, is based on the graph topology, $\mathcal{T}$, vertex-to-vertex relation, $\Re(u, v)$, and other relevant information about the graph, $\mathcal{I}$.

$$\mathcal{H}(u) = \{\mathcal{T}, \Re(u,v), \mathcal{I}\} \tag{9}$$

In a blockchain network, the graph topology $\mathcal{T}$ depends on the type of blockchain. The node types may be transactions, EOA addresses, or smart contracts. The $\Re(u, v)$ represents the relation between different or the same types of nodes. The $\mathcal{I}$ represents the properties of the nodes and edges.

Graph embedding methods can belong to one of three categories: 1) factorisation, 2) random walk, and 3) deep learning. In this work, the random-walk-based graph embedding approach GraphSAGE [35] was chosen to calculate the graph embedding vector of the graphs stated in subsection V-B. The

GraphSAGE samples a tree rooted at each node by recursively expanding the root node's neighbourhood with a fixed number of iterations i. For each iteration, it computes the root node's hidden representation by hierarchically aggregating intermediate nodes' representation from bottom to top. Equation (10) details the calculation for node u intermediate embedding vector h_u^i at i^{th} iteration.

$$h_u^i = \ell(W^i.\mu(h_u^{i-1}, \{h_u^{i-1} \forall u \in \hat{N}(u)\})) \tag{10}$$

$$h_u^0 = X_u \tag{11}$$

where X_u is the initial feature vector of root node u, ℓ is a loss function, W^i is a weight matrix for each convolution layer, μ is an aggregation function, and $\hat{N}(u)$ is a random sample of the root node u's neighbors [36].

In this work, the inputs for GraphSAGE are as follows:

$$\begin{aligned} \mathcal{T} &\leftarrow G_t \text{ or } G_m \\ i &\leftarrow 5 \\ X_u &\leftarrow \mathcal{F} \\ \mu &\leftarrow \text{max pooling} \end{aligned}$$

The G_t and G_m are the Bitcoin transaction and the Ethereum MFT graphs respectively. Here, the number of iterations $i = 5$, and the aggregation function is max pooling. Max pooling calculates the largest value in each patch of a feature vector. At the end of a given number of iterations, GraphSAGE provides an embedding vector $\mathcal{H}(u)$ for the node u. This embedding vector is considered the graph-embedding feature of the node u (transaction or EOA) during the classification. The time complexity for calculating graph embedding feature vector is $O(|V| * D * L * M)$. Sampling neighbours for each node in each layer takes $O(|V| * M)$ and aggregating information from neighbours in each layer takes $O(|V| * D)$ time. Here $|V|$ number of nodes in graph G, D is the dimensionality of the embedding vector, L is the number of layers in GraphSAGE, and M is the number of samples per node in each layer.

VI. Feature evaluation via classification results

This section presents the validation process for engineered structural, embedding, and network property-based features. The features are validated by analysing their significance in classifying both normal and malicious financial activities related to Bitcoin and Ethereum EOA transactions.

A. Experimental setup

The experimental setup utilised three different datasets 1) the normal and ransomware settlement-related Bitcoin transactions, 2) malicious (hacks, theft, and Silk Road-related trades) Bitcoin transactions, and 3) normal and phishing settlement-related Ethereum transactions. Feature evaluation utilised two types of approaches such as 5-fold cross-validation and Grid search (optimised hyperparameters). 5-fold cross-validation was learned on the whole dataset and each individual (Structural, Network, and Embedding) and combined (Structural+Embedding, and Structural+Network+Embedding) feature set was considered for the evaluation. For the Grid search approach, data samples were divided into training (70%) and testing (30%) sets. The Grid search performed cross-validation on the training set for different models of classifiers and identified the best model based on the evaluation results. Finally, the best-trained model for each classifier is tested using a separate testing dataset. The capabilities of the features were evaluated using accuracy(ACC), ROC-AUC(roc-auc), precision (pre), recall (rec), and F1-score(f1) measures. In this experiment, Logistic Regression (LR) [37], Random Forest (RF) [38], Decision Tree (DT) [39], k-Nearest Neighbor (k-NN) [40], Naïve Bayes (NB) [41], XG-boost (XG) [42], Silas [43], and AutoKeras [44] are used to classify transactions (Bitcoin) and EOAs (Ethereum) to identify the suspicious behaviours.

The experiments ran on a computer with Ubuntu 20.04.4 LTS, 11th Gen Intel(R) Core(TM) i7-11850H and 32.0 GB RAM. The code is written in Python, and the graph database is GraphQL [1].

B. Classifiers for blockchain transactions

Logistic regression is a standard classifier for supervised learning. In this experiment, the LR is considered a baseline for classifying malicious transactions and EOA. Further, it is relatively easy to implement and computationally less intensive than other models. The other classifiers, RF, DT, and XG, perform classification by building various decision trees and applying the bagging or boosting idea. These classifiers are computationally more intensive than the LR approach. Silas [43] is a generic data mining and predictive analytics framework that deals with structured data. Also, it contains several modules to understand the model's classification using the XAI technique. The k-NN algorithm is relatively a simple, supervised machine learning algorithm. It works by finding the distances between a target node and all the other nodes in the blockchain dataset to select the specified number of instances (k) closest to the target. It is used to obtain an understanding of the models and the complexity of the data in the classification. AutoKeras [44] is an efficient neural architecture search (NAS) with network morphism. It automatically finds the best combination of data preparation, model, and model hyperparameters for a predictive modelling problem.

C. Validation for structural and embedding features of Bitcoin transactions: Normal and ransomware settlement-related classification

The binary class classification of Bitcoin transactions used the normal and ransomware settlement-related dataset explained in subsection IV-A. Where Class 1 represents ransomware transactions, and Class 0 represents normal transactions. Table III shows the number of training and testing Bitcoin transaction nodes.

Tables IVa, and IVb present classification results for structural, embedding, and structural+embedding engineered features, respectively. Evaluation results in Table IVa were obtained from 5-fold cross-validation, while those in Table IVb

[1] https://graphql.org/

TABLE III: Number of samples in binary class Bitcoin transaction dataset.

Transaction type	# Training	# Testing
Normal	10482	2644
Malicious	5818	1431

were obtained from the test dataset. For the classification results based on 5-fold cross-validation, RF received the best accuracy of 88.66% and roc-auc of 95.04% for the structural features. Silas obtained the best precision of 100.0% for embedding features, recall of 88.59%, and F1-score of 86.36% for structural features. Whereas the results obtained for the test dataset XG obtained the best accuracy of 88.0% and roc-auc of 94.79% for the structural+embedding features. Silas obtained the best precision of 100.0% for embedding features. NB obtained a recall of 99.06% for embedding features and an F1-score of 82.02% for structural+embedding features. In Table IVb, the AutoKeras classifier, only accuracy result is accessible, and no other values are available to compute results for additional evaluation measures of the test dataset.

The accuracy of the structural features in Tables IVa is almost comparable to the accuracy of the structural+embedding features. These results conclude that the structural features greatly influence the binary class classification of Bitcoin transactions. Also, the classifiers based on DT and k-NN obtained a test accuracy of over 81.00%.

D. Validation for structural and embedding features of Bitcoin transactions: Multiclass malicious classification

The multiclass classification of Bitcoin transactions used multiclass malicious activities related datasets described in subsection IV-A. This dataset only contains malicious transactions and they belong to three categories: Class 1 (hacks), Class 2 (theft), and Class 3 (1st and 2nd FBI Silk Road seizures). Table V shows the number of training and testing Bitcoin multi-class transaction nodes. This dataset faced an imbalanced problem. Therefore, the Synthetic Minority Oversampling Technique (SMOTE) was employed to balance the data belonging to the malicious classes. The minority malicious class belonged to the hack attack-based transactions.

Table VIa and VIb represent the performance of various classification models for all three sets of features. The classification results based on 5-fold cross-validation, k-NN received the best accuracy of 95.16% for structural+embedding features, AutoKeras obtained macro precision of 93.03%, macro recall of 92.64% and macro F1-score of 92.33% for structural features. Whereas the results based on the test dataset, RF obtained the best accuracy of 94.97% for structural features, k-NN obtained macro precision of 68.86%, RF obtained macro recall of 79.09% and macro F1-score of 70.15% for structural+embedding features.

E. Validation for individual features of EOA: Normal and phishing settlement-related classification

This section presents the classification results of EOAs based on the structural, network property-based, and embedding features. Table VII presents the number of EOAs used in training and testing samples. The Ethereum EOA's features were evaluated using the non-phishing and phishing datasets described in subsection IV-B. Class 0 represents normal and Class 1 represents phishing EOA. The classification results based on 5-fold cross-validation presented in Table VIIIa informed, that the RF classifier obtained the best accuracy of 98.69% and roc-auc of 99.82%, XG classifier obtained precision, recall and F1-score of 98.25% for structural features. Whereas the results based on testing data in Table VIIIb show that the Silas classifier obtained an accuracy of 96.08%, roc-auc of 99.27% and recall of 98.10%, the RF classifier obtained a precision of 98.06%, and the XG classifier obtained F1-score of 98.80% for structural features.

F. Validation for combined features of EOA: Normal and phishing settlement-related classification

This section details classification results for structural+embedding, and structural+network property-based+embedding EOA features. The classification results based on 5-fold cross-validation informed, the RF classifier obtained the best accuracy of 98.76%, and roc-auc of 99.80% for structural+network properties-based+embedding features, XG classifier obtained a precision, recall and F1-score of 98.19% for structural+embedding features. Whereas the results based on testing data, the RF classifier obtained the best accuracy of 97.86%, the XG classifier obtained a roc-auc of 99.55%, precision of 97.47%, recall of 98.72%, and F1-score of 98.09% for structural+network property-based+embedding features.

Based on the classification results presented in Tables IV, VI and VIII, the accuracy and roc-auc scores for Bitcoin transactions are influenced by the structural features using RF classifier. Although the majority and minority class-based evaluation scores precision and recall were influenced by embedding features using Silas and NB classifiers, respectively and the F1-score was influenced by the structural features using the Silas classifier. These results informed that the generated embedding features for Bitcoin transactions with 20 dimensions have high false positives and false negative costs. The accuracy and roc-auc scores for multiclass Bitcoin transactions are influenced by the structural+embedding features using the k-NN classifier. The class-based evaluation scores macro precision, macro recall, and macro F1-score are also influenced by the structural+embedding features using the XG classifier. Finally, the accuracy and roc-auc scores for the EOA classification are influenced by the structural+network property-based+embedding features. The class-based evaluation score precision is influenced by structural features using the k-NN classifier, recall is influenced by the structural+embedding features using the DT classifier, and the F1-score is influenced by the structural features using the XG classifier. In summary, the decision tree-based classifiers performed well in classification, the accuracy and roc-auc measures are highly dependent on the structural features, and the structural+embedding features are highly contributed to the class-based measures.

TABLE IV: Classification results for the binary class Bitcoin transactions dataset. The numeric columns are the results for structural, embedding and structural+embedding features, respectively. The results are presented via accuracy (acc), ROC-AUC (roc-auc), precision (pre), recall (rec), and F1-score (f1). The "-" indicates that there are no relevant results available from the classifier.

(a) 5-fold cross-validation.

Classifier	Structural					Embedding					Structural+Embedding				
	acc	roc-auc	pre	rec	f1	acc	roc-auc	pre	rec	f1	acc	roc-auc	pre	rec	f1
LR	0.7018	0.7863	0.6706	0.6591	0.6621	0.7015	0.6918	0.3522	0.4999	0.3767	0.7087	0.7996	0.6804	0.6691	0.6723
RF	**0.8866**	**0.9504**	0.8281	0.8284	0.8282	**0.7178**	0.7030	0.7899	0.5026	0.3824	**0.8815**	0.9465	0.8342	0.8344	0.8343
k-NN	0.8533	0.9035	0.7845	0.7848	0.7847	0.6834	0.6227	0.7787	**0.7879**	**0.7813**	0.8490	0.9155	0.7828	0.7924	0.7854
DT	0.8653	0.8519	0.7988	0.7986	0.7987	0.7183	0.7034	0.7448	0.5029	0.3834	0.8611	0.8468	0.8037	0.8104	0.8063
NB	0.8653	0.8519	0.6755	0.6833	0.6737	0.5961	0.6509	0.6144	0.5079	0.3075	0.6650	0.7202	0.6144	0.5079	0.3075
XG	0.8509	0.9134	0.8355	0.8392	0.8371	0.7136	0.7097	0.7863	0.5026	0.3825	0.8510	0.9150	0.8411	0.8463	0.8434
Silas	0.8847	0.9502	**0.8425**	**0.8859**	**0.8636**	0.7077	**0.7061**	**1.0000**	0.5991	0.7493	0.8819	**0.9491**	**0.8439**	**0.8825**	**0.8628**
AutoKeras	0.4013	0.5000	0.2006	0.5000	0.2864	0.4012	0.5000	0.7007	0.5001	0.2864	0.4067	0.5045	0.6937	0.5045	0.2967

(b) Optimised hyperparameters.

Classifier	Structural					Embedding					Structural+Embedding				
	acc	roc-auc	pre	rec	f1	acc	roc-auc	pre	rec	f1	acc	roc-auc	pre	rec	f1
LR	0.7212	0.7939	0.6196	0.5145	0.5621	0.7119	0.7124	0.0000	0.0000	0.0000	0.7244	0.8073	0.6345	0.5367	0.5815
RF	0.8577	0.9331	0.7767	0.7835	0.7801	**0.7245**	0.7244	0.7619	0.0073	0.0145	0.8623	0.9363	0.7850	0.7938	0.7894
k-NN	0.8510	0.9254	0.7819	0.7815	0.7817	0.7193	0.7156	**1.0000**	0.0055	0.0109	0.8319	0.9056	0.7756	0.7878	0.7817
DT	0.8189	0.8019	0.7544	0.7707	0.7624	0.7239	0.7239	0.5000	0.0071	0.0140	0.8282	0.8094	0.7544	0.7815	0.7677
NB	0.7266	0.7662	0.5849	0.7219	0.6463	0.6533	0.6646	0.4019	**0.9906**	0.5719	0.6763	0.7450	0.5725	**0.8852**	0.6953
XG	**0.8675**	**0.9428**	0.7335	**0.8503**	0.7876	0.7244	0.7240	0.5088	0.0067	0.0131	**0.8800**	**0.9479**	0.7309	0.8517	0.7867
Silas	0.8250	0.9107	**0.8782**	0.7483	**0.8080**	0.6699	**0.7246**	**1.0000**	0.6028	**0.7522**	0.8260	0.9129	**0.8808**	0.7685	**0.8208**
AutoKeras	0.7737	-	-	-	-	0.6513	-	-	-	-	0.7386	-	-	-	

TABLE V: Number of samples in the multi-class Bitcoin transaction dataset.

Transaction type	# Training	# Testing
Hacks	32	10
Theft	782	234
Silk road	674	253

G. Class-based performance measure for blockchain transactions

The analysis of classification performance in various classes employs the top-performing classifiers, which achieved the highest test accuracy as in Tables IVb, VIb, and VIIId. The importance of features in each class is analysed by presenting confusion matrices in Fig. 2. Furthermore, Fig. 2a demonstrates how the structural features of Bitcoin transactions exert influence by attaining high true positives, and true negatives, and minimising false positives and false negatives (Table IVb). Additionally, the combination of structural and graph embedding features significantly impacts the achievement of high true positives and true negatives for Class 1 (hacks) and 2 (theft), while classifying most of Class 3 (dark market-related) malicious activity as Class 2, as depicted in Fig. 2b (Table VIb). Similarly, the combination of structural, network property-based, and embedding features exhibits substantial influence, resulting in high true positives, true negatives, and reduced false positives and false negatives, as illustrated in Fig. 2c (Table VIIId). Here true positive: is the number of normal transactions or EOAs correctly identified as normal, true negative: is the number of malicious transactions or EOAs identified as malicious, false positive: is the number of malicious transactions or EOAs identified as normal, and false negative: is the number of normal transactions or EOAs identified as malicious. Based on the observation overall, for both Bitcoin and Ethereum datasets, all three categories of engineered features (structural, network property-based, and embedding) bring promising results in classification accuracy and roc-auc. Among all three, structural features have the most significant impact on classification results.

The highlighted results in bold in Tables IVa, IVb, VIa, VIb, VIIIa, VIIIb, VIIIc, and VIIId show the best results obtained for the classification accuracy and the roc-auc.

VII. Feature importance analysis using Explainable AI

Identifying important features helps to eliminate unimportant features and improve the classification accuracy and performance of the machine learning models. This section analyses the importance of the features for the performance of the best classifiers using the eXplainable Artificial Intelligence (XAI) technique SHAP values. The XAI is a research field that aims to make AI systems and their results more understandable for humans [45]. XAI tools are crucial for understanding and interpreting the decisions made by machine learning models [46]. This research work utilises the well-known game

TABLE VI: Classification results for the multi-class malicious Bitcoin transactions dataset. The numeric columns indicate results for structural, embedding and structural+embedding features, respectively. The results are presented via accuracy (acc), ROC-AUC (roc- auc), macro precision (mar-pre), macro recall (mar-rec), and macro F1-score (mar-f1). The "-" indicates that there are no relevant results available from the classifier.

(a) 5-fold cross-validation.

Classifier	Structural				Embedding				Structural+Embedding			
	acc	mar-pre	mar-rec	mar-f1	acc	mar-pre	mar-rec	mar-f1	acc	mar-pre	mar-rec	mar-f1
LR	0.7349	0.6836	0.6911	0.6842	0.6743	0.5896	0.3752	0.2587	0.8438	0.7184	0.7211	0.7134
RF	**0.9370**	0.9079	0.9051	0.9029	0.6961	**0.6526**	**0.3798**	0.2659	0.9479	0.9094	0.9068	0.9048
k-NN	0.9346	0.8798	0.8727	0.8689	0.6731	0.5361	0.3464	0.2632	**0.9516**	0.8800	0.8750	0.8722
DT	0.9274	0.8712	0.8726	0.8712	0.8679	0.6026	0.3776	0.2622	0.9346	0.8892	0.8879	0.8859
NB	0.6525	0.5929	0.5935	0.5761	0.5012	0.3876	0.3575	0.2265	0.7191	0.6333	0.3634	0.2323
XG	0.8664	0.9135	0.9114	0.9095	0.6488	0.6122	0.3767	0.2604	0.8606	**0.9170**	**0.9150**	**0.9133**
Silas	0.9206	-	-	-	**0.7485**	-	-	-	0.9161	-	-	-
AutoKeras	0.9309	**0.9303**	**0.9264**	**0.9233**	0.3947	0.6502	0.3793	**0.2698**	0.9019	0.9041	0.8945	0.8901

(b) Optimised hyperparameters.

Classifier	Structural				Embedding				Structural+Embedding			
	acc	mar-pre	mar-rec	mar-f1	acc	mar-pre	mar-rec	mar-f1	acc	mar-pre	mar-rec	mar-f1
LR	0.8356	0.6405	0.7291	0.6512	0.7253	**0.4240**	**0.3542**	**0.2619**	0.8723	0.6507	0.7638	0.6637
RF	**0.9497**	**0.6834**	**0.7448**	**0.6957**	0.7369	0.3157	0.3171	0.0446	**0.9478**	0.6821	**0.7909**	**0.7015**
k-NN	0.9419	0.6818	0.6933	0.6865	0.7079	0.4237	0.3528	0.2613	0.9419	**0.6886**	0.6935	0.6909
DT	0.9207	0.6541	0.6831	0.6411	0.7369	**0.4240**	**0.3542**	**0.2619**	0.9323	0.6411	0.6607	0.6399
NB	0.6286	0.5642	0.6094	0.5162	0.5589	0.1697	0.3333	0.2249	0.6828	0.1697	0.3333	0.2249
XG	0.9110	0.6489	0.6494	0.6445	0.7524	0.2658	0.2570	0.0531	0.9226	0.6441	0.6481	0.6432
Silas	0.9399	-	-	-	0.7373	-	-	-	0.9399	-	-	-
AutoKeras	0.9226	-	-	-	**0.7562**	-	-	-	0.9264	-	-	-

TABLE VII: Number of samples in the binary class Ethereum EOA dataset.

Transaction type	# Training	# Testing
Normal	455	112
Malicious	464	118

theory-based feature importance measure SHAP (SHapley Additive exPlanations) [47] to identify the influential features of Bitcoin and Ethereum transactions. The game represents the outcome (normal or malicious), and the players are the features (structural, embedding, and network). It quantifies the contribution that each feature brings to the classification outcome. The outcome of each possible combination of features is considered in determining the importance of a single feature. Based on the classification results presented in Section VI, the RF classifier obtained promising results in classifying malicious behaviour of blockchain transactions. Fig. 3 represents the top six important features of the Bitcoin transaction dataset identified by SHAP based on the outcome of the RF classifier. The min-input and outDegree are identified as important features for only normal and malicious transactions, respectively. The max-input, avg-input, max-output, and min-output are identified as the other important features for both normal and malicious transactions. Significantly, these four features obtained the highest score for malicious transactions. Fig. 4, and 5 partly justify the reasons for the outcome of the feature importance score. Fig. 4 shows that the probability density of maximum, minimum, and average input values for malicious transactions is much higher than that of normal transactions below 225 BTC. Similarly, Fig. 5 shows that the probability density of maximum, minimum, and average output values of malicious, transactions is much higher than that of normal transactions below 600 BTC.

For the Ethereum EOA, Fig. 6, however, shows that there are no common important features between normal and malicious transactions. Five statistical measures of fee, spent amount, and received amount are important features of normal transactions. Whereas, the statistical features of fees and frequency of receiving contribute to the important features of malicious transactions.

The outcome of the feature importance analysis concludes that these engineered features are the most important to train classification models and this correlates with the findings in Section VI. The nature of these important features within a blockchain network may also be used to investigate the underlying behaviour patterns of suspicious transactions.

VIII. Discussion

In the context of this research, significant contributions are made by introducing unified mechanisms designed for

TABLE VIII: Classification results for the binary class Ethereum EOAs dataset. The numeric columns indicate results for individual ((a) and (b)) and combined features ((c) and (d)). The results are presented via accuracy (acc), ROC-AUC (roc-auc), precision (pre), recall (rec), and F1-score (f1). The "-" indicates that there are no relevant results available from the classifier.

(a) 5-fold cross-validation.

Classifier	**Structural**					**Network**					**Embedding**				
	acc	roc-auc	pre	rec	f1	acc	roc-auc	pre	rec	f1	acc	roc-auc	pre	rec	f1
LR	0.9521	0.9943	0.9264	0.9262	0.9261	0.5165	0.5717	0.6338	0.6189	0.6079	0.7217	0.7494	0.8161	0.8160	0.8160
RF	**0.9869**	**0.9982**	0.9788	0.9787	0.9787	0.6446	0.6784	**0.6708**	**0.6702**	**0.6699**	**0.7391**	**0.8300**	0.8426	0.8413	0.8423
k-NN	0.9434	0.9827	0.9242	0.9224	0.9223	0.6322	0.6801	0.6074	0.5932	0.5794	**0.7391**	0.7113	0.8410	0.8423	**0.8472**
DT	0.9695	0.9693	0.9649	0.9649	0.9649	0.6528	0.6564	0.6477	0.6476	0.6477	**0.7391**	0.8286	0.8168	0.8166	0.8166
NB	0.9696	0.9693	0.7418	0.7190	0.7123	0.5868	0.6346	0.6232	0.5907	0.5619	0.6260	0.7589	0.7850	0.7841	0.7839
XG	0.9661	0.9892	**0.9825**	**0.9825**	**0.9825**	**0.6545**	0.6568	0.6671	0.6665	0.6661	0.7363	0.8036	**0.8469**	0.8467	0.8467
Silas	0.9651	0.9931	0.9577	0.9444	0.9412	0.6267	**0.7076**	**0.6944**	0.6173	0.6536	0.7377	0.7958	0.7260	**0.9298**	0.8154
AutoKeras	0.5235	0.5000	0.2616	0.5000	0.3436	0.5235	0.5000	0.2618	0.5000	0.3436	0.5235	0.5000	0.2328	0.5000	0.3177

(b) Optimised hyperparameters.

Classifier	**Structural**					**Network**					**Embedding**				
	acc	roc-auc	pre	rec	f1	acc	roc-auc	pre	rec	f1	acc	roc-auc	pre	rec	f1
LR	0.9279	0.9834	0.9321	0.9679	0.9497	0.4957	0.5329	0.5488	0.7564	0.6361	0.6271	0.7025	**0.7938**	0.8141	0.8038
RF	0.9576	0.9913	**0.9806**	0.9744	0.9775	0.6239	0.6329	**0.6437**	0.7179	0.6788	0.7203	0.7770	0.7514	0.8526	0.7988
k-NN	0.9491	0.9895	0.8862	0.9487	0.9164	**0.6282**	0.6381	0.6341	0.6667	0.6499	**0.7245**	0.7723	0.7740	**0.8782**	**0.8228**
DT	0.9449	0.9449	0.9682	0.9744	0.9712	0.5940	0.5997	0.6419	0.6667	0.6541	0.7161	0.7742	0.7514	0.8333	0.7903
NB	0.9110	0.9442	0.7881	0.5962	0.6788	0.5619	0.5513	0.5556	**0.9295**	**0.6954**	0.5974	0.7142	0.7471	0.8141	0.7791
XG	0.9491	0.9880	0.9745	0.9808	**0.9880**	0.5897	0.6218	0.5909	0.6667	0.6265	0.7161	**0.7889**	0.7746	0.8589	0.8146
Silas	**0.9608**	**0.9927**	0.9451	**0.9810**	0.9627	0.5000	**0.6391**	0.3109	0.7391	0.4378	0.4869	0.7667	0.7256	0.8207	0.7702
AutoKeras	0.9522	-	-	-	-	0.5478	-	-	-	-	0.5130	-	-	-	-

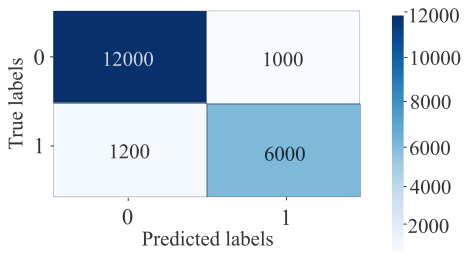
(c) 5-fold cross-validation.

Classifier	**Structural+Embedding**					**Structural+Network+Embedding**				
	acc	roc-auc	pre	rec	f1	acc	roc-auc	pre	rec	f1
LR	0.9565	0.9946	0.9339	0.9337	0.9337	0.9380	0.9878	0.9333	0.9330	0.9330
RF	**0.9869**	**0.9979**	0.9733	0.9731	0.9731	**0.9876**	**0.9980**	0.9753	0.9749	0.9749
k-NN	0.9391	0.9808	0.9036	0.9011	0.9009	0.9421	0.9849	0.9035	0.9011	0.9009
DT	0.9695	0.9692	0.9651	0.9649	0.9649	0.9710	0.9712	0.9700	0.9699	0.9699
NB	0.9130	0.9462	0.7949	0.7947	0.7947	0.8636	0.9021	0.7931	0.7928	0.7928
XG	0.9888	0.9652	**0.9819**	**0.9819**	**0.9819**	0.9678	0.9882	**0.9813**	**0.9812**	**0.9812**
Silas	0.9619	0.9946	0.9315	0.9315	0.9315	0.9695	0.9935	0.9729	0.9114	0.9411
AutoKeras	0.5235	0.4656	0.5000	0.2328	0.5000	0.3177	-	-	-	-

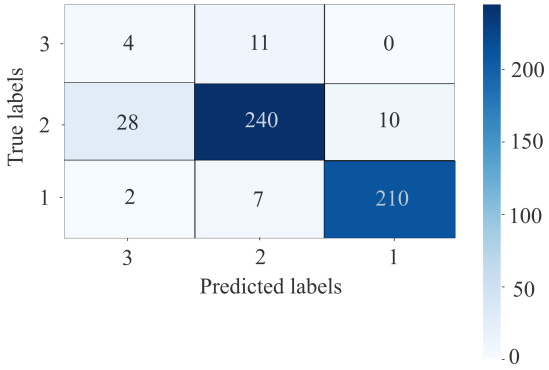
(d) Optimised hyperparameters.

Classifier	**Structural+Embedding**					**Structural+Network+Embedding**				
	acc	roc-auc	pre	rec	f1	acc	roc-auc	pre	rec	f1
LR	0.9194	0.9838	0.9207	0.9679	0.9438	0.9658	0.9921	0.9207	0.9679	0.9438
RF	**0.9576**	**0.9919**	0.9683	0.9807	0.9745	**0.9786**	0.9952	0.9745	0.9808	0.9776
k-NN	0.9449	0.9909	0.8286	0.9294	0.8761	0.9487	0.9882	0.8286	0.9295	0.8761
DT	0.9449	0.9449	0.9451	**0.9936**	0.9688	0.9615	0.9615	0.9615	0.9615	0.9615
NB	0.8602	0.9005	0.7500	0.8269	0.7866	0.8419	0.8948	0.7443	0.8397	0.7892
XG	0.9533	0.9873	**0.9746**	0.9872	**0.9809**	0.9573	**0.9955**	**0.9747**	**0.9872**	**0.9809**
Silas	0.9565	0.9913	0.9390	0.9809	0.9595	0.9478	0.9888	0.9451	0.9810	0.9627
AutoKeras	0.5478	-	-	-	-	0.9522	-	-	-	-

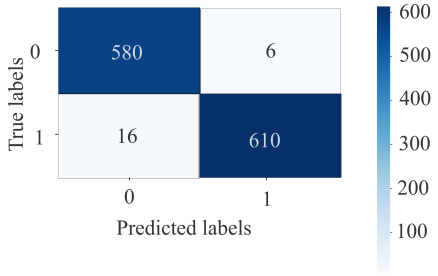
the extraction of three distinct categories of features from raw blockchain transactions (structural) and graphs of the blockchain networks (network properties and embedding). The effectiveness of these features was validated by classifying



(a) The class-based performance for binary class Bitcoin transactions with structural+embedding features.



(b) The class-based performance for multi-class malicious Bitcoin transactions with structural+embedding features.



(c) The class-based performance for binary class Ethereum EOAs with structural+network+embedding features.

Fig. 2: Confusion matrix for blockchain transactions.

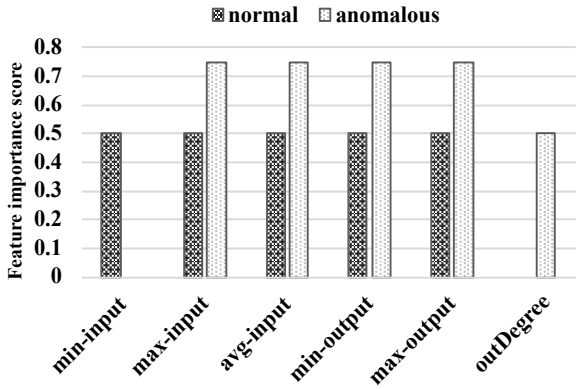


Fig. 3: Top six features influencing the classification of binary class Bitcoin transactions.

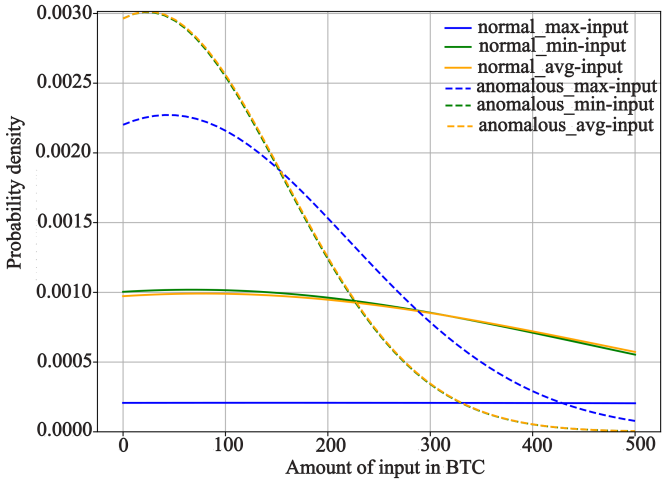


Fig. 4: Probability density of statistical measures for input value feature of binary class Bitcoin transactions.

normal and malicious activities in the blockchain network.

In contrast to the datasets mentioned in the literature, such as [12] and [25], the Bitcoin and Ethereum normal and malicious transactions data utilised in this study possess balanced samples in each class. This characteristic facilitates the learning of generalised patterns by models and enhances their efficiency in detection. The graph-based features engineered via this proposed work capture the temporal dynamics of the ransomware and phishing settlement-based transactions. The significance of these features is reflected in the combined features classification results presented in Tables IVb, and 'Structural+Embedding' column in VIb, VIIIb and VIIId. The feature importance analysis using XAI reveals that the structural features of Bitcoin transactions influenced the identification of malicious transactions. In contrast, the combination of structural and embedding features made a significant contribution to the identification of malicious Bitcoin transactions and Ethereum EOA.

Table IX presents a comparison between the results achieved for binary class Bitcoin transactions and Ethereum EOAs using combined features and the results reported in the literature. For direct comparison purposes, the experimental results from this research are rounded to three decimal places. The term Proposed (S+E) refers to Structural+Embedding for Bitcoin and Proposed (S+N+E) refers to Structural+Network+Embedding for Ethereum. The results for the classification of binary class Bitcoin transactions are referenced from Table IVa. For Ethereum, related work [25] extracted features using trans2Vec and then used one-class SVM for classification. For the comparison, this work conducted a separate classification using one-class SVM for combined features of EOA and then compared the results with the results reported in the literature.

The datasets examined in the related studies exhibit a substantial imbalance issue, particularly with a high number of

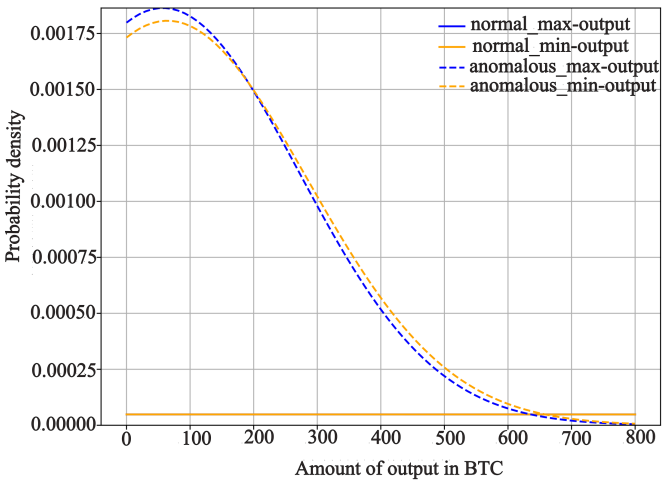


Fig. 5: Probability density of statistical measures for output value feature of binary class Bitcoin transactions.

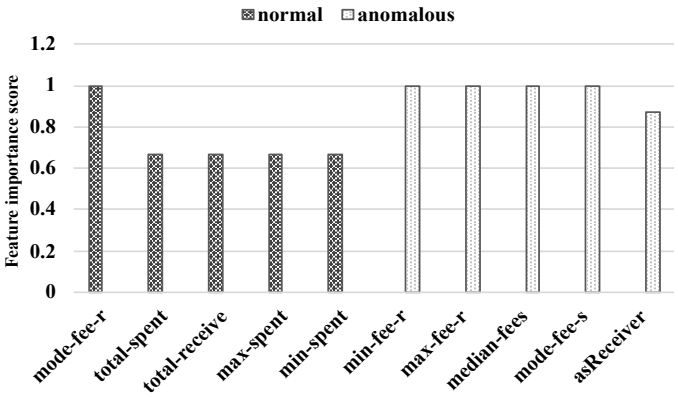


Fig. 6: Top five features influencing the classification of binary class Ethereum EOAs.

positive samples which is reflected in their reported precision results. However, the features derived from this research yield significantly improved recall and F1-score results when compared with the outcomes reported in the literature. The combined features identified via this work support classification models to achieve high recall which implies that these features are sensitive to the presence of positive instances, and it is successful in avoiding a significant number of false negatives. A high F1-score indicates a balance between precision and recall, implying that these engineered features perform well in accurately identifying both false positives and false negatives. The marginal enhancement in the F1-score of research works [13] and [14] informs the limitation in the dimension of the embedding vector. In general, the Bitcoin transaction graph involves multiple hops, while the Ethereum MFT graph is limited to a maximum of two hops (account-based model). Consequently, employing a 20-dimensional embedding proves to be more effective for Ethereum than Bitcoin, as evidenced by the 1.0 precision (high false positive) of embedding features. This limitation contributes to only marginal

TABLE IX: Comparison details for the classification results obtained via engineered features and the results in related literature. Where columns precision (pre), recall (rec), and F1-score (f1) result in the classification. The "-" indicates that there are no relevant results available in the literature.

Blockchain	Classifier	Dataset	pre	rec	f1
Bitcoin	RF	Elliptic [12]	0.971	0.675	0.796
		Elliptic++ [14]	0.975	0.719	0.828
		HBTBD [15]	0.789	0.609	0.687
		MoneyLaundering [13]	-	-	0.830
		Proposed (S+E)	0.834	0.834	0.834
Ethereum	One-class SVM [26]	Xblock [25]	0.927	0.893	0.908
		Proposed (S+N+E)	0.915	0.962	0.938

improvements in the F1-score for the Structural+Embedding features of Bitcoin transactions.

Upon analysing the outcomes, it becomes evident that engineered features in this work yield significantly better recall and F1-score results when compared to previous analyses on both Bitcoin and Ethereum datasets.

Limitations: The proposed approach has a number of limitations such as the APIs from various platforms providing transaction data in different formats; this will change the reference name of the key features in the generalised graph modelling algorithm. Another limitation is the dimension of embedding was limited to 20. Also, the number of hidden layers for GraphSAGE with Neo4j can only be two fixed-sized hidden layers. However, the exhaustive feature collection resulting from the existing data sources of the Bitcoin wallet addresses and Ethereum EOAs for financial crime-based analysis contributes to advancing blockchain analysis by deepening the understanding and enabling benchmarking.

IX. Conclusion

This work proposed a unified feature engineering pipeline to extract three categories of features for malicious activity detection in Bitcoin and Ethereum blockchain networks. The XAI-based feature importance analysis revealed that for both Bitcoin and the Ethereum networks the statistical measures-based features yield higher predictive performance in the classification of malicious actors. The comparison outcomes indicate that, when considering the proposed feature sets, the Structural+Network+Embedding features demonstrate superior performance in classifying EOAs. On the other hand, the use of a 20-dimensional embedding vector marginally enhances the performance for classifying Bitcoin transactions when using Structural+Embedding features. In future research, the identified features will serve as the foundation for training blockchain-specific classification models, conducting suspicious community identification, and exploring graph-based node properties. Additionally, the effectiveness of the key components, namely, graph modelling and embedding generation, will be evaluated on real-time blockchain activity.

References

[1] W. Dai, C. Dai, K.-K. R. Choo, C. Cui, D. Zou, and H. Jin, "Sdte: A secure blockchain-based data trading ecosystem," *IEEE Transactions*

on Information Forensics and Security, vol. 15, pp. 725–737, 2019. [Online]. Available: https://doi.org/10.1109/tifs.2019.2928256

[2] N. B. Truong, K. Sun, G. M. Lee, and Y. Guo, "Gdpr-compliant personal data management: A blockchain-based solution," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1746–1761, 2019. [Online]. Available: https://doi.org/10.1109/tifs.2019.2948287

[3] J. Chen, C. Wang, Z. Zhao, K. Chen, R. Du, and G.-J. Ahn, "Uncovering the face of android ransomware: Characterization and real-time detection," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 5, pp. 1286–1300, 2017. [Online]. Available: https://doi.org/10.1109/tifs.2017.2787905

[4] "Cerber ransomware campaign," https://www.zdnet.com/article/how-bitcoin-helped-fuel-an-explosion-in-ransomware-attacks/, 2021.

[5] C. Akcora, "Bitcoinheist: Topological data analysis for ransomware prediction on the bitcoin blockchain," in *IJCAI*, 2020. [Online]. Available: https://doi.org/10.24963/ijcai.2020/612

[6] R. Zhang, G. Zhang, L. Liu, C. Wang, and S. Wan, "Anomaly detection in bitcoin information networks with multi-constrained meta path," *Journal of Systems Architecture*, vol. 110, p. 101829, 2020. [Online]. Available: https://doi.org/10.1016/j.sysarc.2020.101829

[7] P. Nerurkar, S. Bhirud, D. Patel, R. Ludinard, Y. Busnel, and S. Kumari, "Supervised learning model for identifying illegal activities in bitcoin," *Applied Intelligence*, vol. 51, no. 6, pp. 3824–3843, 2021. [Online]. Available: https://doi.org/10.1007/s10489-020-02048-w

[8] Z. Wu, J. Liu, J. Wu, Z. Zheng, and T. Chen, "Tracer: Scalable graph-based transaction tracing for account-based blockchain trading systems," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 2609–2621, 2023.

[9] P. Nerurkar, Y. Busnel, R. Ludinard, K. Shah, S. Bhirud, and D. Patel, "Detecting illicit entities in bitcoin using supervised learning of ensemble decision trees," in *ICICM*, 2020, pp. 25–30. [Online]. Available: https://doi.org/10.1145/3418981.3418984

[10] C. G. Akcora, Y. Li, Y. R. Gel, and M. Kantarcioglu, "Bitcoinheist: Topological data analysis for ransomware detection on the bitcoin blockchain," *arXiv preprint [Web Link]*, 2019.

[11] "BitcoinHeistRansomwareAddressDataset," UCI Machine Learning Repository, 2020, DOI: https://doi.org/10.24432/C5BG8V.

[12] M. Weber, G. Domeniconi, J. Chen, D. K. I. Weidele, C. Bellei, T. Robinson, and C. E. Leiserson, "Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics," *arXiv preprint arXiv:1908.02591*, 2019.

[13] J. Lorenz, M. I. Silva, D. Aparício, J. T. Ascensão, and P. Bizarro, "Machine learning methods to detect money laundering in the bitcoin blockchain in the presence of label scarcity," in *Proceedings of the First ACM International Conference on AI in Finance*, 2020, pp. 1–8. [Online]. Available: https://doi.org/10.1145/3383455.3422549

[14] Y. Elmougy and L. Liu, "Demystifying fraudulent transactions and illicit nodes in the bitcoin network for financial forensics," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, ser. KDD '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 3979–3990. [Online]. Available: https://doi.org/10.1145/3580305.3599803

[15] J. Song and Y. Gu, "Hbtbd: A heterogeneous bitcoin transaction behavior dataset for anti-money laundering," *Applied Sciences*, vol. 13, no. 15, 2023. [Online]. Available: https://www.mdpi.com/2076-3417/13/15/8766

[16] P. Monamo, V. Marivate, and B. Twala, "Unsupervised learning for robust bitcoin fraud detection," in *ISSA*. IEEE, 2016, pp. 129–134. [Online]. Available: https://doi.org/10.1109/issa.2016.7802939

[17] T. Pham and S. Lee, "Anomaly detection in the bitcoin system - a network perspective," Nov 2016.

[18] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: identifying density-based local outliers," in *COMAD*, 2000, pp. 93–104. [Online]. Available: https://doi.org/10.1145/335191.335388

[19] B. Podgorelec, M. Turkanović, and S. Karakatič, "A machine learning-based method for automated blockchain transaction signing including personalized anomaly detection," *Sensors*, vol. 20, no. 1, p. 147, 2020. [Online]. Available: https://doi.org/10.3390/s20010147

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

[21] D. Kondor, M. Pósfai, I. Csabai, and G. Vattay, "Do the rich get richer? an empirical analysis of the bitcoin transaction network," *PLOS ONE*, vol. 9, no. 2, pp. 1–10, 02 2014. [Online]. Available: https://doi.org/10.1371/journal.pone.0086197

[22] F. Scicchitano, A. Liguori, M. Guarascio, E. Ritacco, and G. Manco, "A deep learning approach for detecting security attacks on blockchain." in *ITASEC*, 2020, pp. 212–222.

[23] "Ethereum transaction dataset," https://www.kaggle.com/bigquery/crypto-ethereum-classic, 2021.

[24] S. T. Jeyakumar, A. C. Eugene Yugarajah, Z. Hóu, and V. Muthukkumarasamy, "Detecting malicious blockchain transactions using graph neural networks," in *Distributed Ledger Technology*, N. Dong, B. Pillai, G. Bai, and M. Utting, Eds. Singapore: Springer Nature Singapore, 2024, pp. 55–71.

[25] J. Wu, Q. Yuan, D. Lin, W. You, W. Chen, C. Chen, and Z. Zheng, "Who are the phishers? phishing scam detection on ethereum via network embedding," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2020. [Online]. Available: https://doi.org/10.1109/tsmc.2020.3016821

[26] K.-L. Li, H.-K. Huang, S.-F. Tian, and W. Xu, "Improving one-class svm for anomaly detection," in *Proceedings of the 2003 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.03EX693)*, vol. 5, 2003, pp. 3077–3081 Vol.5.

[27] J. S. Tharani, E. Y. A. Charles, Z. Hóu, M. Palaniswami, and V. Muthukkumarasamy, "Graph based visualisation techniques for analysis of blockchain transactions," in *2021 IEEE 46th Conference on Local Computer Networks (LCN)*. IEEE, 2021, pp. 427–430.

[28] "Blockchain data API," https://www.blockchain.com/api/blockchain_api, 2021.

[29] "Ethereum transaction dataset," https://api.blockcypher.com/v1/eth/main/txs/, 2022.

[30] B. K. Mohanta, D. Jena, S. S. Panda, and S. Sobhanayak, "Blockchain technology: A survey on applications and security privacy challenges," *Internet of Things*, vol. 8, p. 100107, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2542660518300702

[31] O. Shafiq, "Bitcoin hacked transactions 2010-2013," 2019. [Online]. Available: https://dx.doi.org/10.21227/7f0c-df28

[32] C. Lee, S. Maharjan, K. Ko, and J. W.-K. Hong, "Toward detecting illegal transactions on bitcoin using machine-learning methods," in *International Conference on Blockchain and Trustworthy Systems*. Springer, 2019, pp. 520–533. [Online]. Available: https://doi.org/10.1007/978-981-15-2777-7_42

[33] H. Baek, J. Oh, C. Y. Kim, and K. Lee, "A model for detecting cryptocurrency transactions with discernible purpose," in *2019 Eleventh International Conference on Ubiquitous and Future Networks (ICUFN)*. IEEE, 2019, pp. 713–717. [Online]. Available: https://doi.org/10.1109/icufn.2019.8806126

[34] C. F. Negre, U. N. Morzan, H. P. Hendrickson, R. Pal, G. P. Lisi, J. P. Loria, I. Rivalta, J. Ho, and V. S. Batista, "Eigenvector centrality for characterization of protein allosteric pathways," *National Academy of Sciences*, vol. 115, no. 52, pp. E12 201–E12 208, 2018. [Online]. Available: https://doi.org/10.1073/pnas.1810452115

[35] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *NeurIPS*, 2017, pp. 1025–1035.

[36] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 1, pp. 4–24, 2020. [Online]. Available: https://doi.org/10.1109/tnnls.2020.2978386

[37] M. P. LaValley, "Logistic regression," *Circulation*, vol. 117, no. 18, pp. 2395–2399, 2008.

[38] S. J. Rigatti, "Random forest," *Journal of Insurance Medicine*, vol. 47, no. 1, pp. 31–39, 2017.

[39] Y.-Y. Song and L. Ying, "Decision tree methods: applications for classification and prediction," *Shanghai archives of psychiatry*, vol. 27, no. 2, p. 130, 2015.

[40] L. E. Peterson, "K-nearest neighbor," *Scholarpedia*, vol. 4, no. 2, p. 1883, 2009.

[41] I. Rish *et al.*, "An empirical study of the naive bayes classifier," in *IJCAI 2001 workshop on empirical methods in artificial intelligence*, vol. 3, no. 22, 2001, pp. 41–46.

[42] X. Yu, J. Zhou, M. Zhao, C. Yi, Q. Duan, W. Zhou, and J. Li, "Exploiting xg boost for predicting enhancer-promoter interactions," *Current Bioinformatics*, vol. 15, no. 9, pp. 1036–1045, 2020.

[43] H. Bride, C.-H. Cai, J. Dong, J. S. Dong, Z. Hóu, S. Mirjalili, and J. Sun, "Silas: A high-performance machine learning foundation for logical reasoning and verification," *ESWA*, vol. 176, 2021. [Online]. Available: https://doi.org/10.1016/j.eswa.2021.114806

[44] H. Jin, Q. Song, and X. Hu, "Auto-keras: An efficient neural architecture search system," in *Proceedings of the 25th ACM SIGKDD KDD*, 2019, pp. 1946–1956. [Online]. Available: https://doi.org/10.1145/3292500.3330648

[45] M. van Lent, W. Fisher, and M. Mancuso, "An explainable artificial intelligence system for small-unit tactical behavior," in *Proceedings of*

the 16th Conference on Innovative Applications of Artifical Intelligence, ser. IAAI'04. AAAI Press, 2004, p. 900–907.

[46] R. Dwivedi, D. Dave, H. Naik, S. Singhal, R. Omer, P. Patel, B. Qian, Z. Wen, T. Shah, G. Morgan, and R. Ranjan, "Explainable ai (xai): Core ideas, techniques, and solutions," *ACM Comput. Surv.*, vol. 55, no. 9, jan 2023. [Online]. Available: https://doi.org/10.1145/3561048

[47] M. Sundararajan and A. Najmi, "The many shapley values for model explanation," 2020.

Jeyakumar Samantha Thanrani is a PhD candidate at the School of Information and Communication Technology at Griffth University, Gold Coast, Australia. Her research interests are Blockchain Technology, Cyber Security, Graph modelling, and Machine Learning.

Eugene Yugarajah Andrew Charles is a Senior Lecturer at the Department of Computer Science, University of Jaffna, Sri Lanka. He obtained his Ph.D. from the School of Engineering at Cardiff University, UK. His research interests lie in machine learning and natural language processing.

Zhe Hou is a Senior Lecturer at Griffith University, Australia. He obtained his Ph.D. degree from the Australian National University in 2015. His research mainly focuses on automated reasoning, formal methods, autonomous systems, machine learning, and blockchain.

Punit Rathore is an Assistant Professor at the Indian Institute of Science, India in the Robert Bosch Centre for Cyberphysical Systems, jointly with the Centre for Infrastructure, Sustainable Transportation, and Urban Planning. He obtained his PhD at the University of Melbourne, Australia in 2019. His research interests are in big data analytics, unsupervised learning, explainable ML, and data-driven analytics for smart cities, including intelligent transportation systems.

Marimuthu Palaniswami is a Fellow of the Institute of Electrical and Electronic Engineering (IEEE). He currently serves as a Professor at the University of Melbourne, Australia. He earned his M.E. degree in Electrical, Electronic, and Control Engineering from the Indian Institute of Science, and his Ph.D. from the University of Newcastle, Australia. His research interests include sensor networks, the Internet of Things (IoT), machine learning, pattern recognition, and signal processing and control.

Vallipuram Muthukkumarasamy received a PhD from Cambridge University, England, and a B.Sc.Eng. (Hons.) from the University of Peradeniya, Sri Lanka. He is currently attached to the School of ICT, Griffith University, Australia, as Professor. His research areas include cyber security, blockchain technology, and wireless sensor networks.