International Journal of Advanced Technology and Engineering Exploration, Vol 10(103)
ISSN (Print): 2394-5443 ISSN (Online): 2394-7454
http://dx.doi.org/10.19101/IJATEE.2022.10100578

Research Article

An interpretable ensemble model framework for real-time anomaly detection and prediction of Ethereum blockchain transactions

Sabri Hisham*, Mokhairi Makhtar and Azwa Abdul Aziz
Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin, 22000 Besut, Terengganu, Malaysia

Received: 24-December-2022; Revised: 25-June-2023; Accepted: 26-June-2023

Abstract
The blockchain ecosystem is often referred to as a technology that ensures security. However, there have been concerns in the real world regarding the security of blockchain applications, like what happens with conventional database systems. The anonymous design of blockchain provides cyber-attackers with opportunities to commit crimes, resulting in an increase in scams, phishing, code manipulation of smart contracts, Ponzi schemes, and other fraudulent activities. Consequently, many individuals and national economies worldwide have suffered significant losses. Detecting fraudulent behavior in blockchain transactions manually is infeasible due to the enormous amount of data involved. Therefore, the optimal method for identifying abnormalities within the blockchain network is to combine a blockchain platform with a machine learning approach. This study employs filter method techniques such as mutual information (MI), analysis of variance (ANOVA), and recursive feature elimination (RFE) to identify the ideal set of features based on the maximum accuracy value, considering the feature dimension (k value). The study screens and ranks the top 10 feature sets using the feature importance random forest (RF) classifier, based on the dataset produced by the best filter approach (yielding higher accuracy). Subsequently, an ensemble methodology is used to create the final model, utilizing the final dataset consisting of 10 features. The purpose of this approach is to enhance the level of anomaly detection in the blockchain network. To determine the effectiveness of the proposed model, experiments are conducted, comparing it against individual classifiers such as extreme gradient boosting (XGB), decision tree (DT), logistic regression (LR), random forest (RF), and k-nearest neighbor (KNN). The study's findings reveal that the ensemble voting approach achieves a 96.78% accuracy rate, surpassing the accuracy of the individual classifier models that utilize optimal features. Additionally, the study's findings suggest that the selection of features and their quantity significantly impact the output of the model.

Keywords
Ethereum, Blockchain, Features extraction, Ensemble method, Anomaly detection.

1.Introduction
Blockchain is a growing technology nowadays. This technology is often associated with the terms "distributed" and "decentralized" technology that does not require a central authority (CA) to control certain processes [1]. Therefore, the acceptance of this technology is increasing with the introduction of some of the latest applications in the world of decentralized financial (DeFi), non-fungible tokens (NFT), and metaverse. Previously, blockchain technology was widely adopted in several popular domains, such as digital government [2], health [3], business management [4], and smart cities [5]. The understanding of blockchain is linked to several main functions: public key, private key, hash value, and peer-to-peer (P2P) network.

In principle, blockchain operates based on generating different keys (public, private key) for the encryption and decryption process through cryptographic mechanisms. On the other hand, this key differs from the key method generated through the symmetric cryptographic method, which is guided by using the same key in the encryption and decryption process [6].

In contrast, the hash value is generated from a hash generator consisting of several digit numbers with a fixed length. Therefore, the hash value is a unique id for blockchain application development. Technically, the same data input produces the same hash value. However, even a slight change in the data input will produce a different hash value [7]. The foundation of blockchain architecture is based on the concept of a P2P network [8]. Basically, this architecture allows sharing elements such as files, tasks, or work to be executed between several peers. Peers refer to

*Author for correspondence

machines with internet protocol (IP) addresses, such as notebooks, computers, workstations, servers, and others connected to the blockchain network. Some of these peers are also known as nodes in relation to each other in the P2P network ecosystem [9].

In more detail, the blockchain platform is based on blocks linked to each other for storing several data transactions in each block through a cryptographic mechanism. Therefore, identifying the relationship between the current and previous blocks is done through the identity of the unique block hash address on each block, also known as the "previous block hash". Thus, this block consists of several important elements, such as the block identifier, block header, and Merkle Tree (referring to the data structure that takes care of data integrity). Technically, this blockchain stores past data ledger transactions (not discarded) as historical data and gives an advantage to the development of applications based on the supply chain [10]. In addition, the ledger in the blockchain provides audit trailing value on the collection of past data transactions [11]. Therefore, a ledger in the blockchain network can be compared to a ledger in an accounting system. However, the blockchain ledger is open (no centralized entity for data verification), while the ledger in a conventional accounting system has a centralized concept (control by given entity access). Furthermore, the blockchain architecture allows it to be immune from any form of record modification or traceability [12].

The evolution of blockchain technology started with Bitcoin, which was first introduced in 2009 and is popular in the world of cryptocurrency. Nevertheless, the operation of Bitcoin, a few years after it was identified, has several weaknesses. Thus, this improvement became the basis for the development of Ethereum, which has become the largest blockchain network after Bitcoin. Therefore, this improvement led to the introduction of digital smart contracts, which makes a big difference between Ethereum and Bitcoin. As a result, the adoption of smart contracts in decentralized application (Dapp) development is increasing. It is proven by the introduction of enterprise Dapp applications in various sectors, including insurance, health, automotive, agriculture, banking, and so on [11]. From the point of view of marketing capital, Ethereum contributed more than 20 billion in April 2020[13].

Despite the proliferation of Bitcoin and Ethereum-based applications today, the issue of security crime is a significant challenge. As a result, the increase in cases of scams, phishing, cryptojacking, smart contract bugs, Ponzi schemes, and so on has become a significant threat to Ethereum-based applications [14–16]. This security threat was commented on by the author [17], who stated that permission less blockchain accounts give space to attackers to carry out cyber-criminal activities and illegal behavior. Lack of policy reinforcement has resulted in a 50% increase in cybercrime cases since 2017, and many people have become victims of this fraud [18]. In 2015, there were 13,000 victims of fraud schemes who lost $11 million [19]. In the past century, conventional crimes such as Ponzi schemes have also occurred, and the most recent ones employ the more modern blockchain platform to find victims [14].

Cyber-attacks on the blockchain network can be dealt with earlier if there is a mechanism for detecting abnormalities in transactions in a predictive manner. Thus, service providers can make early preparations as a precaution against cyber-attacks. Therefore, data science and machine learning (ML) can be assimilated with blockchain technology to curb all forms of cybercrime. However, the biggest challenge is that account addresses on the blockchain network are anonymous [20]. As a result, it will be challenging to identify the cause of abnormalities in blockchain transactions. Therefore, studies on the detection of anomalies in blockchain networks do not solely rely on a single technological solution. Instead, deep learning (DL) [21] and artificial intelligence (AI) [22] are used to examine the source code of smart contracts to find anomalies. Analyzing the entire transactions of the dataset extracted from the blockchain network directly is not easy. The size of data that is too large has a significant impact on the analyzer because it involves a high cost of energy resources, long processing time, high hardware specification requirements, and so on [23]. Furthermore, because the manual method is done by humans, it is very prone to analysis errors, data loss, data corruption, and being overlooked. Therefore, automatic processing analysis using an ML approach can speed up the detection of anomalies, save time, reduce the rate of manual errors, and automate the manual detection process.

Nevertheless, most previous research has struggled to determine the optimal feature for developing a more effective anomaly detection mechanism. In ML, feature extraction and selection are very difficult task, especially in detecting fraud or anomalies [24]. Determining the features that generate the highest level of detection is a significant challenge for researchers. The difficulty is compounded when the

Ethereum architecture undergoes frequent enhancements that alter its features. Consequently, the previously manufactured model must be revised, and the redevelopment of the new model must begin from scratch. The selection of features has a significant effect on the overall efficacy of the finalized model. For instance, duplicated or unrelated features can decrease a model's performance [1].

Consequently, a few previous studies have focused solely on a single technique for feature selection based on feature relevance. This method is overly biased toward the selection of randomly determined techniques without identifying other techniques that have the potential to yield good performance on a given number of features (k). The research conducted by [25] employs a method of feature importance extreme gradient boosting (XGB) classifier that, after data preparation, selects the top 10 features from the entire feature set. Using the XGB model, this investigation achieved an accuracy of 96.3%. Meanwhile, a study by [26, 27] also does not use a specific feature selection method and focuses more on manual feature extraction during the data preparation phase for Ethereum account fraud detection.

Based on this scenario, this study aims to investigate and analyze the selection of the most optimal features using multiple filter method techniques (mutual information (MI), analysis of variance (ANOVA), and recursive features elimination (RFE)) based on the feature dimension size (k value). The selection of the optimal filtering method is determined by the precision value derived from the k-dimensional value. Then, based on the ranking order generated by the random forest (RF) method for determining the importance of a feature, the 10 finest features are chosen. The ensemble method is Among the ML techniques that produce high accuracy values. The approach of this method is to produce a stronger classifier from a combination of several weak classifiers [23]. The ensemble approach is proven to lower the error rate on overfitting variants and models [28]. Therefore, the main goal of the study is to produce a high-performance anomaly detection prediction model using the ensemble approach using the best features. Previous research has investigated methods for enhancing model performance by tuning and optimizing parameters or hyperparameters. Nevertheless, this study aims to focus on extracting and selecting the best model-impacting features by modifying the ensemble learning model.

In order to accomplish this purpose, the following procedures must be taken: 1) Access and extract Ethereum transaction datasets (9841,50) from Kaggle.com, which contain a mixture of normal and aberrant transactions (phishing, scams, and so on); 2) Apply the extracted datasets to an ML model; 3) Investigating characteristics, variance, data type, etc. is a component of dataset exploration; 4) Data purification by deleting null-value records, duplicate records, and highly linked characteristics; 5) Data preparation includes data normalization and data set resampling to balance the amount of majority (normal) and minority (abnormal) classes; 6) Implement Best Feature Filter Method; 7) select the top 10 best features; 8) The ensemble voting technique is employed for model construction (training and testing); 9) the accuracy values of various individual classifiers are compared to the ensemble voting model in order to analyze the experimental outcomes. The accuracy of Ensemble Voting, as shown by the experiment, is 96.78%, the greatest among all other models.

The organizational structure of this research paper consists of several sections. Section 2 provides a summary of previous investigations. Section 3 outlines the research methods employed in this study. Section 4 presents the experimental assessment results. Section 5 contains a detailed discussion of the findings. Finally, Section 6 concludes the paper by summarizing the study and outlining potential future work.

2.Literature review

Anomaly detection is the process of identifying unusual behavior that deviates from the normal pattern in a dataset [29–31]. This methodology is utilized in different domains of data science, data mining, and data analytics. Anomalies are seen as two main types: angles, which are strange signs that appear when damage or problems occur in the network and behavior that appears after being detected in data patterns. Consequently, the practice of anomaly detection is utilized in numerous technological fields. Among these is the creation of the most recent ensemble model framework for the identification of anomalies in Blockchain-based internet of things (IoT) devices [32]. This is because IoT devices are extremely susceptible to security vulnerabilities. Using a collection of IoT Raspberry Pie simulators, this project effectively identified numerous forms of threats.

The objective of a study by [25] is to distinguish between licit and illicit accounts on the Ethereum network using the top 10 features generated by the feature importance technique (XGB). This study examined transactions from legal accounts (2,502) and illegal accounts (2,179). This data set was compiled from two primary sources (EtherscamDB and Geth Client). Using the XGB classifier, the results of this investigation achieved a detection accuracy of 96.3%. For the convenience of future researchers, this study's data set is shared publicly on the GitHub repository platform. However, this study only employs a single feature importance technique (XGB) without performing a comparative analysis of various techniques (producing the best features) based on the feature dimension value (k).

A study by [27] analyzed illicit Ethereum accounts using a data set designated (fraud, non-fraud) with a smaller number of features (6) compared to (42) when using a data set from Kaggle. The correlation coefficient analysis method is utilized to determine the finest features for dimension size reduction. This study has effectively demonstrated that using both selected and complete model features improve performance. In addition to analyzing Ethereum transactions, [24] analyzed smart contracts using the Ponzi scheme dataset generated by [14, 33]. This data set refers to the extraction of the bytecode data set (1904 contract address) using a web crawler. It comprises a subset of normal contracts (1596) and abnormal contracts (308). This study presents hybrid features analysis, which combines characteristics such as operation code (OPCODE), source code, and transaction behavior. Using feature vector techniques, this hybrid method creates a new data set based on a combination of features (feature extraction). In addition, the feature importance technique is used to select the top 10 features for the purpose of developing anomalous models. The ensemble voting technique utilized in the proposed model effectively produced a higher level of accuracy than the research conducted by [14, 33].

A study was conducted to perform anomaly detection on a dataset of 10,000 smart contracts, aiming to identify the behavioral patterns within different contract categories, including games, exchange, finance, social, wagering, and high-risk [34]. Using the data-slicing technique, this analysis was able to identify fourteen essential characteristics of smart contracts. This dataset was utilized to train and validate the long short-term memory (LSTM) network model. The experimental results demonstrate that this model is effective at detecting anomalies in smart contracts. Smartpol tools have been developed by [35] to detect anomalies in smart contracts using a ML approach. This study was developed using a real smart contract data set (49512) extracted from Etherscan. The pre-processing phase involves transforming words or phrases into vector number format using natural language processing (NLP) techniques (Word2Vec, NGram, and FastText). The feature extraction phase involves the use of two main methods: particle swarm optimization (PSO) and gravitational search algorithms (GSA). The development of the classification model uses transductive support vector machines (TSVM), and the study results show that this tool can produce an accuracy value of more than 96%.

Smart contract source code analysis through semantic and syntactic methods has been carried out by [36] using a data set (5,000 smart contracts) produced from a study [37]. This source code analysis involves several combinations of techniques, such as control flow graph (CFG), vectorization, and feature extraction (TextCNN). The results of experiments conducted based on five types of vulnerabilities (implicit visibility, implicit visibility, integer underflow, time dependency, and reentrancy) succeeded in producing precision (96%) and recall (90%) values. The contract source code analysis technique is also implemented semantically by [38] using the smart contract dataset (149,363) extracted from the XBlock [39] platform and the multi-task learning method. The first step entails the extraction of feature vectors for semantic analysis based on contract input. The second step is the development of a model using a convolutional neural network (CNN) by extracting training and testing data from shared layers. As a consequence of experimental observation, this multitasking model demonstrates superior performance, time savings, and a lower cost than single-task models.

A study by [40] has analyzed phishing addresses based on randomly extracted network node transactions (1259 phishing address labels). This study has introduced trans2vec (network embedding algorithm) to extract phishing addresses that have been identified. Trans2vec consists of two main processes: feature extraction (DeepWalk and node2vec methods) and detection model development (one-class SVM). The study results show that this model can detect phishing and normal nodes more effectively. A Ponzi scheme detection study in the blockchain network was conducted by [41] using a data set labeled Ponzi scheme based on graph analysis (homogeneous and

heterogeneous transaction graphs). This Ponzi detection solution is named heterogeneous feature augmentation (HFAug), which is carried out through two experiments, namely HFAug (manual feature and graph random walks) and HFAug (graph neural network). The results of the study show improved detection on the dataset and help existing Ponzi detection techniques produce better results.

Using the decision tree (DT) ensemble technique,[42] conducted a study on the detection of Bitcoin entities with anomalous characteristics. This experiment applies nine features to a dataset extracted directly from the Bitcoin network. Through a comparison of ensemble proposed models with individual classifier models (RF and DT), the study's findings indicate that the recommendation model accurately detects 66% of illegally active users. A study analyzing the effectiveness of feature engineering practice (feature selection) to improve classification was done by [1]. This experiment was conducted using datasets from the blockchain using two types of ensemble techniques, namely ensemble stacking and ensemble boosting. These two ensemble models were compared with several individual classifier models such as support vector machine (SVM), k-nearest neighbors (KNN), logistic regression (LR), DT, and multilayer perceptron (MLP). The results of the study show that the ensemble model produces a better level of anomaly detection effectiveness compared to individual classifiers.

From the point of view of network devices, they are also not exempt from security threats. Thus,[43] has developed a blockchain-empowered ensemble anomaly detection (BCEAD) solution for threat detection on wireless sensor networks (WSN). This experiment was conducted using the isolation forest (IF) model using the KDD CUP'99 dataset to ensure its effectiveness. The results of the study prove that BCEAD successfully produces anomaly detection performance that has been improved through the experiments that have been carried out. The study of unlawful behaviors in Bitcoin transactions was conducted by [44] utilizing the Bitcoin dataset (1216 accounts) that was categorized into 16 types. This work compared ensemble models (RF, XGB) to individual classifiers such as SVM and LR). The results of the study indicate that the ensemble model is more accurate than the individual classifier model for predicting when criminal actions would be detected in Bitcoin.

A comparative analysis was conducted to evaluate supervised learning models for the purpose of identifying fraudulent activity within blockchain networks [45]. This experiment compared eight distinct types of supervised learning models (LR, MLP, naïve Bayes (NB), adaptive boosting (AdaBoost), DT, SVM, RF, and deep neural network (DNN)). AdaBoost, RF, and SVM provide the greatest experimental results compared to other models. A study of the effectiveness of the ensemble learning technique was conducted by [46] using the Ethereum dataset to identify the characteristics of anomalous transactions. In the experiment conducted by comparing ensemble stacking models with individual classifier variations (LR, SVM, RF, AdaBoost), The Stacking Ensemble Learning model gave a better accuracy value than other classifiers. Ponzi schemes are a type of cybercrime that is growing more popular in the field of blockchain technologies. Thus, researchers [47] have developed the Ponzitech model, which detects anomalies of Ponzi behavior on the Ethereum network. Using data analysis gathered from Google BigQuery, our study identified 532 Ponzi schemes successfully. The Ponzitect model has successfully acquired the highest F-Score (98%) among all other models.

Based on the behavior of accounts in Ethereum transactions,[26] investigates the detection of legitimate and unlawful activity. This work developed detection models using a dataset including 4,681 rows of data and three model types (RF, KNN, and XGB). Experimental data demonstrate that XGB achieves greater average accuracy (96.80%) than competing models. This study does not, however, employ a particular method of feature importance for optimal feature selection. Numerous losses have been incurred by the public due to phishing scams. Consequently, [48] conducted the activity detection analysis utilizing a dataset of phishing-labeled accounts from two major portals. This study employs the node2vec approach to classify the dataset's features. The SVM model exhibited the best performance based on the F-score value of 0.846. A smart contract is the logic controller for the use case process on the Ethereum blockchain. Therefore, the creation of solidity-based contract programs is susceptible to errors that result in security breaches. Thus, [49] has conducted the study, analyzing the source code in byte code format to find problems in smart contracts through the development of the DefectChecker tool. This study employs an open-source dataset to evaluate the efficacy of various technologies. The results demonstrated that this tool acquired an F-Score of 88.8% and successfully

identified faults in the total number of smart contracts (levels 1-3) with an F-Score of 88.8%. (25,815).

Based on the analysis of previous studies, most studies use the approach of extracting features through manual analysis methods. There are also studies that only use one filter feature method or feature selection (feature importance) technique that is randomly selected for feature selection. There is no study that shows a comparison of feature selection methods to be analysed and decided experimentally to finalise the most optimal feature selection method. Based on this scenario, this study analyses three main feature selection methods for the selection of the best method that can produce the best performance. Furthermore, this study also integrates two main classifiers (Extra-trees and XGB) through an ensemble voting approach to produce a more comprehensive anomaly detection method. This proposed method is reviewed in more detail in section 3.

3.Methods

The proposed ecosystem framework for anomaly detection in Ethereum transactions using the ensemble learning method is explained in this section (see *Figure 1*). This process begins with the process of collecting Ethereum transaction data. It continues with data exploration, data cleaning, data preparation, data splitting, unbalanced data handling, best feature filter method, feature importance analysis, selecting the top ten best features, and final model development. This study uses Python and Jupyter Notebook as integrated development environments (IDEs) to analyze datasets and model development.

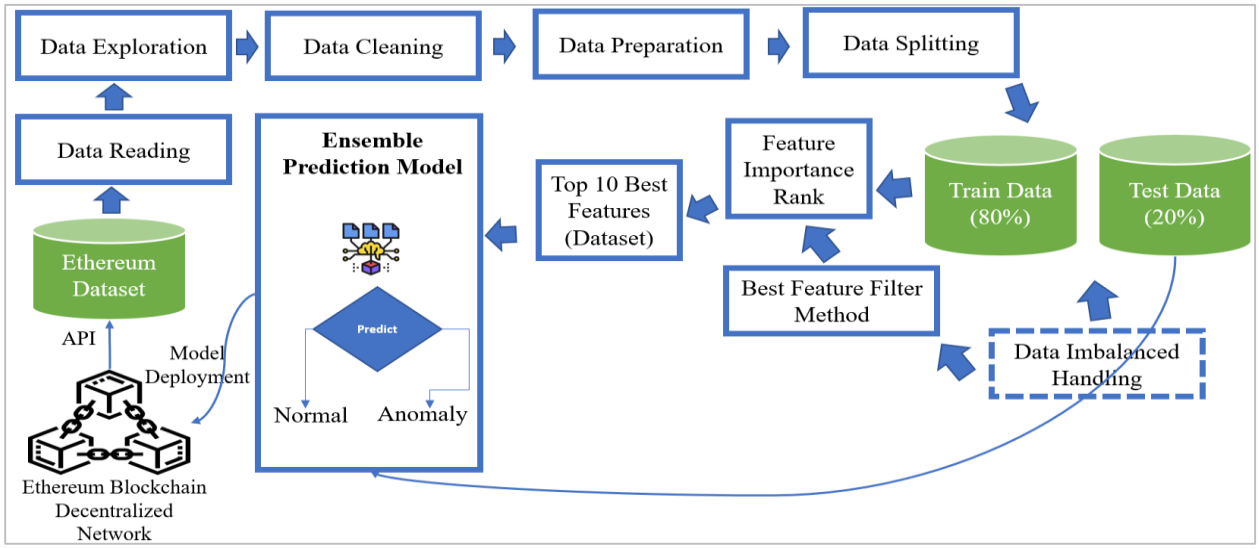

Figure 1 Anomaly detection framework

3.1Data reading

The source of the dataset is one of the most essential factors in the construction of ML models. This study employs an Ethereum network dataset for its analysis. In the Ethereum ecosystem, metadata entities are typically identifiable by their unique public addresses. Consequently, multiple transactions, such as smart contracts, miners, exchanges, tokens, NFT, etc., possess various Ethereum addresses. This address is used for fund transfer transactions and activities [50]. In terms of analysis, investigations on the pattern or behavior of data transactions can be conducted by extracting data from the blockchain network for analysis. This is comparable to the use of data sources from conventional databases. All addresses for normal and abnormal transactions are generally stored on the Ethereum network. Abnormal activities relate to transaction addresses involved in cybercriminal operations like phishing, fraud, cryptojacking, and manipulation of smart contracts.

This aligns with the description in the previous section that explains security threats to the Ethereum network as a result of the cyber-attack activity. Therefore, anomaly detection using an ML approach is crucial for detecting abnormal behavior in blockchain transactions. This study uses the available dataset source of Ethereum transactions taken from kaggle.com's public repository. This Ethereum data transaction (before the cleaning process) is 2.88Mb in

size and contains 9841 transactions and 50 features. This data consists of addresses for all transactions that have been labeled (flagged) as "normal" and "abnormal". Based on the source of the dataset, it consists of 7662 normal transactions and 2179 labeled as abnormal transactions.

3.2Data exploration

The next step is to perform data exploration based on the extracted dataset. Among the initial analysis produced is access to data information, including the list of features (columns), the data type of each feature, and the number of features. The characteristics are examined in terms of numerical attributes (float, integer), categorical, and variance. Referring to the dataset used, some data features (columns) are declared as independent variables (iv) and dependent variables (also known as target variables).

Therefore, the features column "FLAG" is the target attribute, consisting of binary values (0,1). This binary value refers to the transaction labeled "0" to represent a "normal" transaction, while the label "1" is "abnormal". Next, a check on the distribution pattern of transactions with a value of "0" and "1" is carried out based on data that has not gone through the data cleaning process (see *Figure 2*). The result of the distribution shows normal transactions of 77.86% (7662 records) and 22.14% (2179 records) are abnormal transactions.

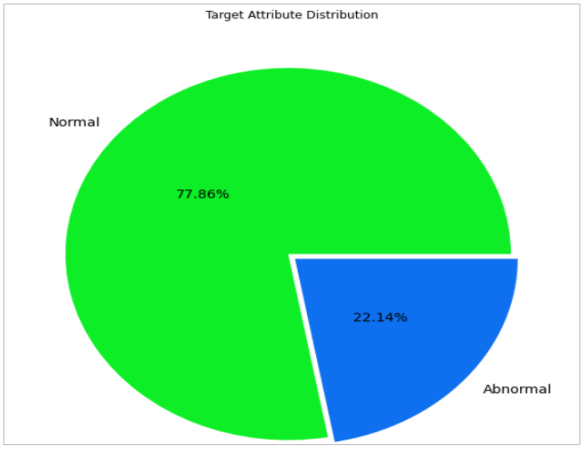

Figure 2 Target attribute distribution

A heatmap diagram (see *Figure 3*) can also be used to look for connections between normal and unusual transactions.

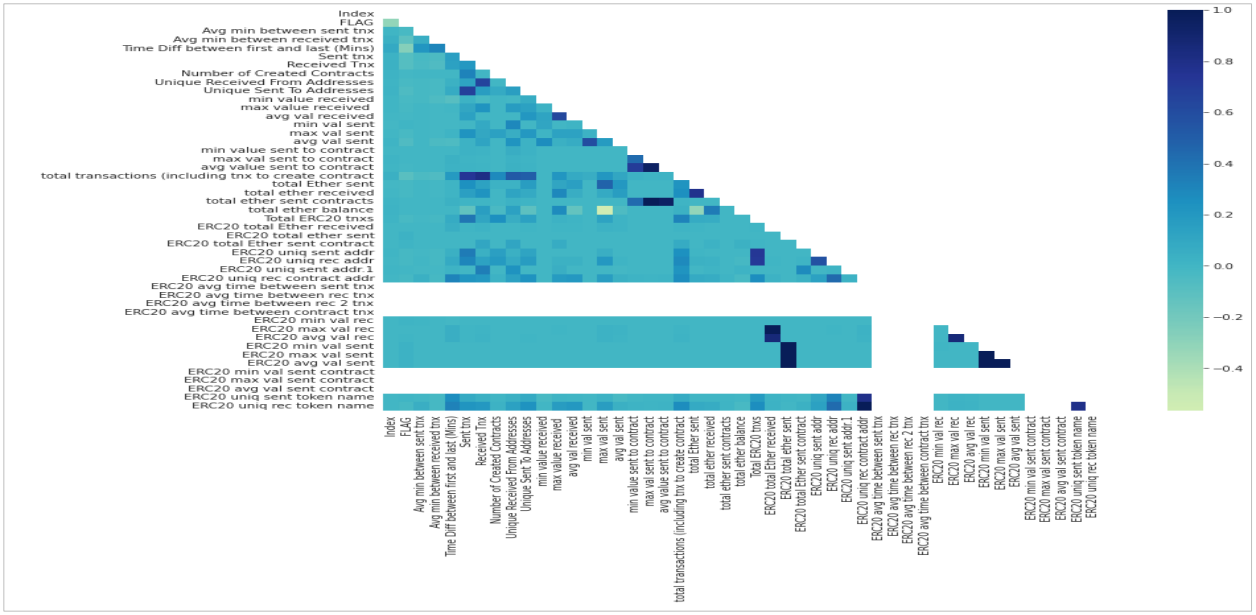

Figure 3 Normal and abnormal correlation matrix

3.3Data cleaning

In this step, a check is made on data rows with a null value (see *Figure 4*) and duplicates. Duplicate records will be discarded, while records with a null value will be replaced with the median value of the numerical attribute (see *Figure 5*). Normally, there is a feature value of 0 that does not contribute to the performance of the final model. Therefore, this dataset is filtered with 0 variant values to be discarded. Apart from that, some features are not related to the development of the model and need to be discarded. This can be seen in the analysis of the frequency of values in two columns

(ERC20mostrectokentype, ERC20mostsenttokentype), which shows that most of the values are 0.

Referring to the correlation matrix in the heatmap diagram, some features need to be discarded because they are highly correlated, and attribute values exceeding 4000 are worth 0 (see *Figure 6).* Among the features involved are "ERC20totalEtherreceived","ERC20totalethersent",ERC20uniqreccontractaddr","ERC20maxvalsent","totalethersentcontracts","maxvalsenttocontract","totalERC20tnxs","ERC20totalEthersentcontract","ERC20uniqsentaddr","ERC20uniqrecaddr","ERC20minvalrec", "ERC20maxvalrec","ERC20minvalcent","ERC20uniqsenttokenname","ERC20uniqrectokenname", "minvaluesenttocontract" and "ERC20uniqsentaddr".

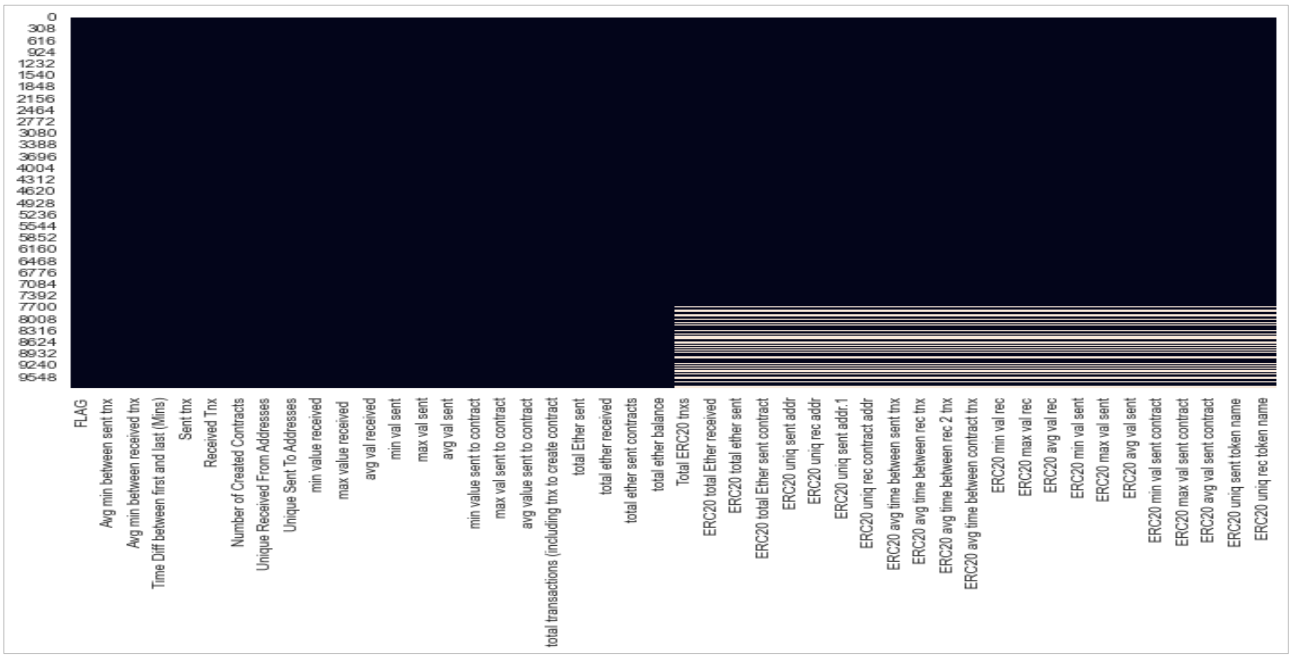

Figure 4 Visualize null pattern

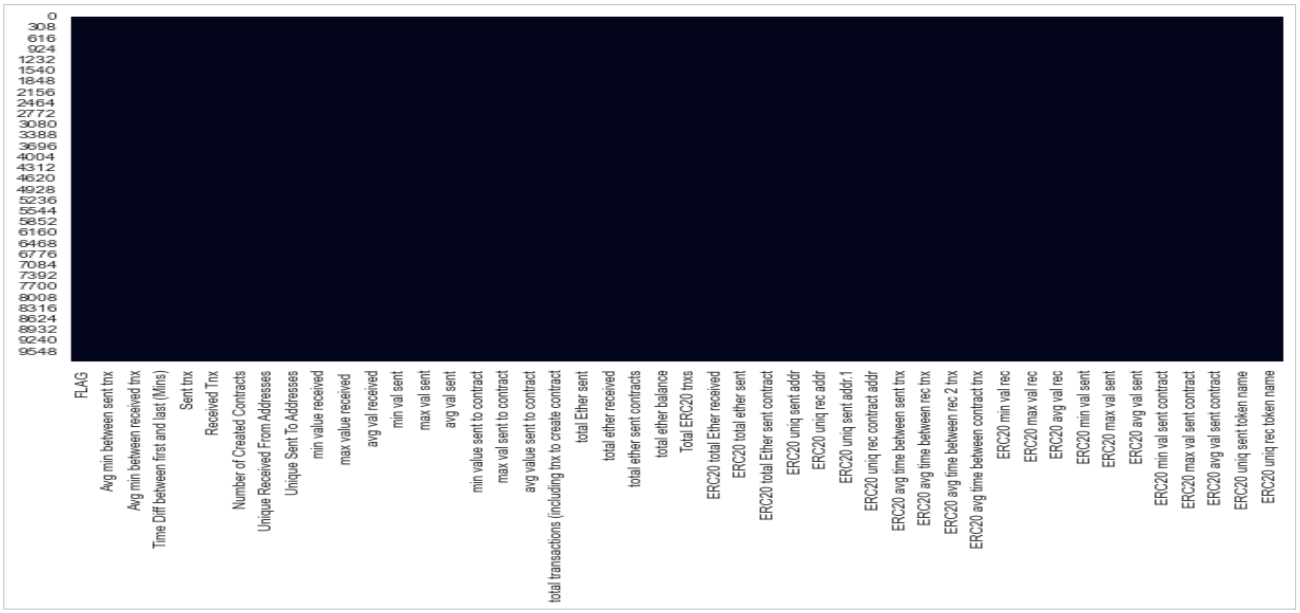

Figure 5 Visualize after median replacement (numerical attribute)

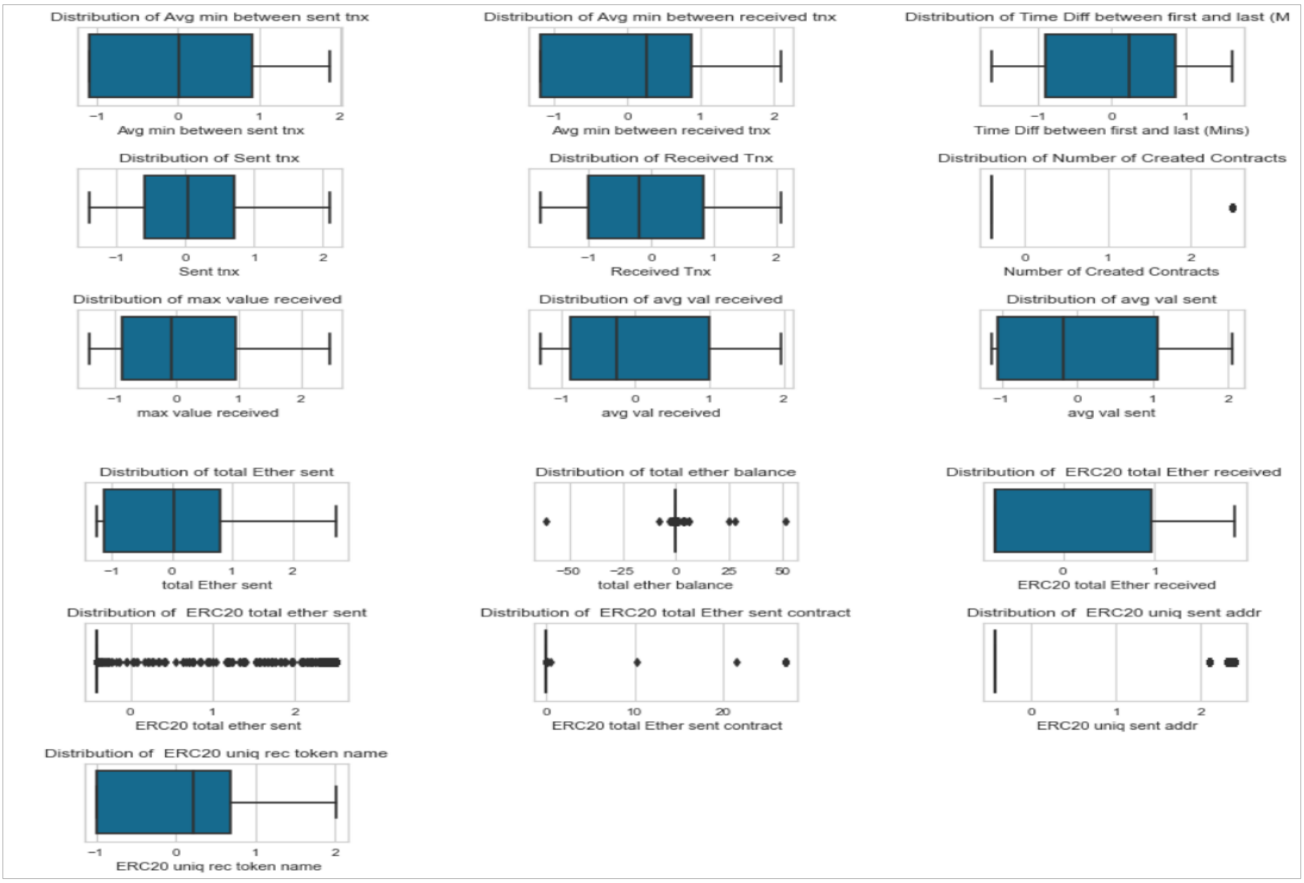

Figure 6 Distribution of features

Table 1 provides a summary of actions conducted on removed characteristics during the data cleansing process. *Table 2* summarises the features that have been updated and the rows that have been eliminated. After the cleaning process is implemented, there is a change in the number of features and rows compared to the original dataset (see *Table 3*). Finally, the model is trained using a total of 17 new features (see *Table 4*).

Table 1 Summary of features that has been removed

Features	Data type	Action	Reason	Total features
Index Address	Numerical Categorical	Filtering the total number of features with many unique values	Many unique values and is used as an identifier	**2**
ERC20mostsenttokentype, ERC20mostrectokentype	Categorical	Filtering the frequency of features with 0 values	The frequency at 0 value is high	**2**
ERC20avgtimebetweensenttnx, ERC20avgtimebetweenrectnx, ERC20avgtimebetweenrec2tnx, ERC20avgtimebetweencontracttnx, ERC20minvalsentcontract, ERC20maxvalsentcontract, ERC20avgvalsentcontract	Numerical	Filtering the features with 0 variance	Drop features with 0 variances. These features will not help in the performance of the model	**7**

Features	Data type	Action	Reason	Total features
Totaltransactions (includingtnxtocreatecontract, totalethersentcontracts, maxvalsenttocontract, ERC20avgvalrec, ERC20avgvalrec, ERC20maxvalrec, ERC20minvalrec, ERC20uniqreccontractaddr, maxvalsent, ERC20avgvalsent, ERC20minvalsent, ERC20maxvalsent, TotalERC20tnxs, avgvaluesenttocontract.UniqueSentToAddresses, UniqueReceivedFromAddresses, totaletherreceived, ERC20uniqsenttokenname, minvaluereceived, minvalsent, ERC20uniqrecaddr	Numerical	Check with the correlation matrix	Highly correlated	**21**
minvaluesenttocontract, ERC20uniqsent addr1	Numerical	Investigate the distribution of our features using boxplots	Some features present a small distribution	**2**
Total number of features				**34**

Table 2 Summary of features and rows that have been updated

Features	Action	Reason
TotalERC20tnxs,ERC20totalEtherreceived,ERC20totaletherse nt,ERC20totalEthersentcontract,ERC20uniqsentaddr,ERC20un iqrecaddr,ERC20uniqsentaddr.1,ERC20uniqreccontractaddr,E RC20avgtimebetweensenttnx,ERC20avgtimebetweenrectnx,E RC20avgtimebetweenrec2tnx,ERC20avgtimebetweencontractt nx,ERC20minvalrec,ERC20maxvalrec,ERC20avgvalrec,ERC2 0minvalsent,ERC20maxvalsent,ERC20avgvalsent,ERC20minv alsentcontract,ERC20maxvalsentcontract,ERC20avgvalsentcon tract,ERC20uniqsenttokenname,ERC20uniqrectokenname,ERC 20mostsenttokentype,ERC20mostrectokentype	Check missing values	Replace the categorical missing values with the mode and numerical values with the median
All features that have duplicate rows	Check duplicated rows	546 rows are duplicates. Drop duplicates rows

Table 3 Summary of the latest dataset

Dataset	Rows (Instances)	Total features	Features		Size of features after cleaning	Size of instances (rows) after cleaning
			Categorical	Numerical		
Ethereum Transaction	9841	50	3	47	17	9295

Table 4 A summary of 17 features of the cleansed dataset

No.	Features	Data type	Data description
1	FLAG	Numerical (int64)	Normal (value: 0) or Abnormal (value: 1) transaction
2	Avgminbetweensenttnx	Numerical (float64)	The average number of minutes between sent transactions for a certain account
3	Avgminbetweenreceivedtnx	Numerical (float64)	The average number of minutes between received transactions for a certain account
4	TimeDiffbetweenfirstandlast (Mins)	Numerical (float64)	The difference in time between the first and final transaction
5	Senttnx	Numerical (int64)	Quantity of typical transactions sent
6	ReceivedTnx	Numerical (int64)	Total number of typical transactions received
7	NumberofCreatedContracts	Numerical (int64)	Total Number of Contract Transactions Created
8	Maxvaluereceived	Numerical (float64)	Most value ever received in Ether
9	Avgvalreceived	Numerical (float64)	The average amount of Ether received

No.	Features	Data type	Data description
10	Agvalsent	Numerical (float64)	The minimum amount of Ether ever transmitted
11	TotalEthersent	Numerical (float64)	The total amount of Ether sent to an address
12	Totaletherbalance	Numerical (float64)	Total Ether Balance after all transactions have been executed.
13	ERC20totalEtherreceived	Numerical (float64)	Total ERC20 token transactions received in Ether
14	ERC20totalethersent	Numerical (float64)	Total number of ERC20token transactions sent in Ether
15	ERC20totalEthersentcontract	Numerical (float64)	Total ERC20 token transfers in Ether to other contracts.
16	ERC20uniqsentaddr	Numerical (fl oat64)	Quantity of ERC20 token transactions transmitted to unique account identifiers
17	ERC20uniqrectokenname	Numerical (float64)	The uniqueness of ERC20 coins received

3.4Data preparation

After going through the cleaning process, the data needs to be divided into two main types: training data and testing data. In this experiment, the training and test data ratio is (80:20). Next, to ensure the performance of the produced model, the normalization transformation process on the training features is carried out. However, the distribution of datasets labeled as normal (majority class) and abnormal (minority class) is unbalanced. Therefore, the synthetic minority oversampling technique (SMOTE) is used to balance the number of normal and abnormal data distributions. *Table 5* shows a comparison of how this data was spread out (normally or not) before and after SMOTE.

Table 5 Target feature distribution (SMOTE, without SMOTE)

Oversampling approach	Normal	Abnormal
SMOTE	**6116**	**6115**
Without SMOTE	1757	6115

3.5Features filtering

The data set that has gone through the data preparation process will determine the best features out of 17 features. Thus, this study used the three feature filtering methods (MI, ANOVA, and RFE). *Figure 7* depicts an analysis of accuracy in relation to the feature dimension size (k) for three varieties of filter methods (x-axis = k, y-axis = accuracy). Three experiments were conducted based on the accuracy rating using the RF classifier for each dimension size of k-features. The best filter method is determined based on the highest accuracy value. The analysis results show that the ANOVA method was selected as the best filter method by producing the highest accuracy value on features (k = 14) with an accuracy of 0.941.

3.6Features importance

ANOVA has been chosen as the optimal filtering technique for generating a new dataset with seventeen features. The RF feature importance method is used to select the 10 finest features from the 17 overall features ordered by importance priority. *Figure 8* depicts the priority analysis of these 17 features. The summary of the top 10 features is shown in *Table 6.*

3.7Ensemble learning

The ensemble technique is one of the dominant approaches in ML. This technique can help improve weaknesses in weak models to produce new models with higher performance. The analogy of this ensemble technique is that if we unite, we will be stronger than if we did something individually. Various approaches exist to implement ensemble learning in the actual world[24]. Based on the labeled data set (0 - normal, 1- abnormal) and the optimal feature size (10 features), two ensemble classifiers, bagging (extra-tree) and boosting XGB, are trained to surmount the shortcomings of individual classifiers. The extra-tree classifier is a bagging learning model (bootstrap aggregation) constructed from random trees and capable of reducing dataset variations. The following describes the Extra Trees classifier formula based on the base learner $C_j(x)$ as shown in Equation 1.

$$extraT(x)=argmin_{i\varepsilon\{0,1\}} \sum_{j=1}^{m} \emptyset\left(c_j(x)=i\right) \quad (1)$$

The XGB classifier is the most recent gradient-boosting-based DT variant. The boosting algorithm is founded on the interaction of k values, as shown in the following Equation 2:

$$f_k(x)=f_{k-1}(x)+\propto G_k(x) \quad (2)$$

The ensemble method is proven to improve the class prediction performance of a combination of different classifiers. The same classifier algorithm can generate different sub-classes. Therefore, in this study, a soft voting ensemble method was proposed for each

classifier to vote on each class (see *Figure 9*). The last class prediction results from the proposed ensemble method based on the weighting of the soft voting mechanism. This proposed model is generally implemented based on three main phases: 1) Dividing the dataset at a ratio (80%, 20%) for model training and testing. 2) two classifiers (Extra-Tree and XGB) were trained using the training dataset, tested, and made predictions using the test dataset.3) This classifier model's test and prediction processes form an ensemble model to produce the final prediction. The process of testing and predicting the ensemble model will determine whether the transaction is normal or abnormal.

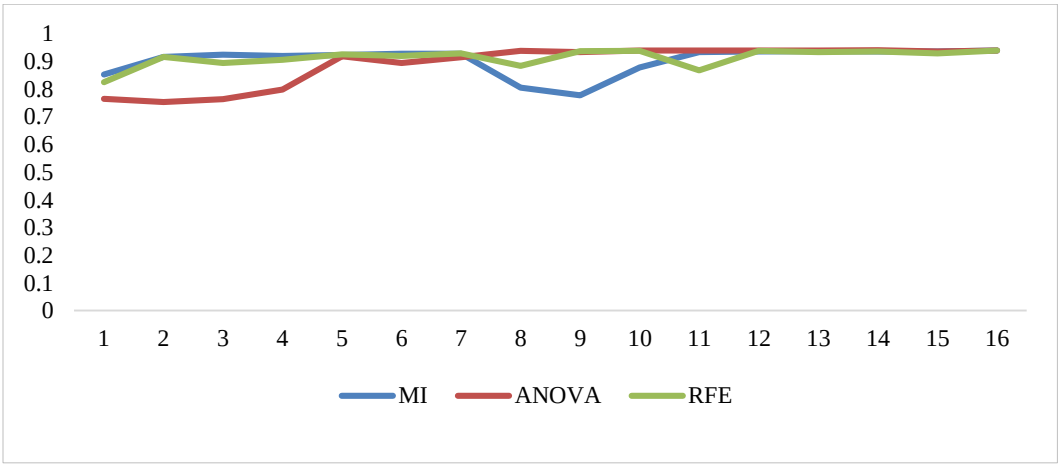

Figure 7 Accuracy analysis based on k-value (MI, ANOVA, and RFE)

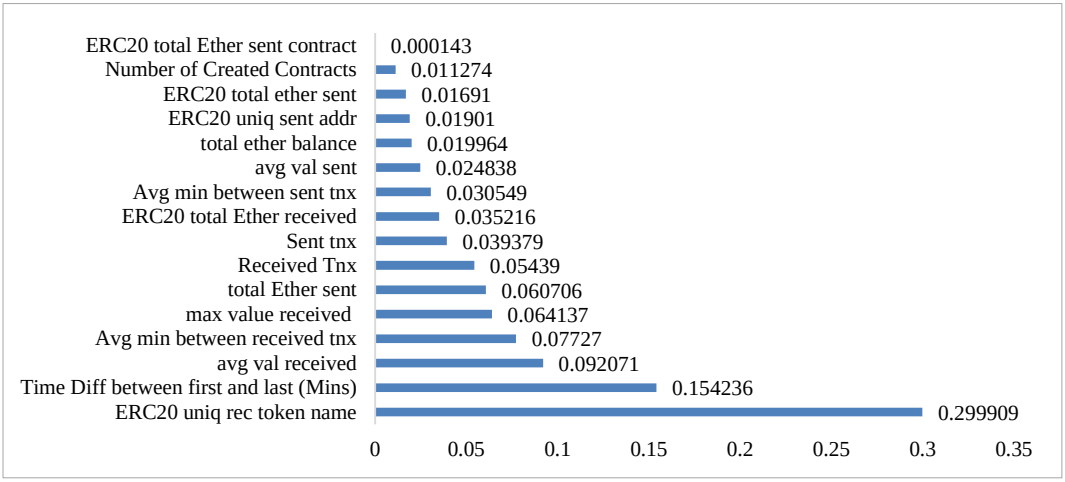

Figure 8 Features importance using RF classifier (17 features)

Table 6 Top 10 best features

No.	Features	Rank
1	ERC20 uniq rec token name	0.299909
2	Time Diff between first and last (Mins)	0.154236
3	avg val received	0.092071
4	Avg min between received tnx	0.07727
5	max value received	0.064137
6	total Ether sent	0.060706
7	Received Tnx	0.05439
8	Sent tnx	0.039379
9	ERC20 total Ether received	0.035216
10	Avg min between sent tnx	0.030549

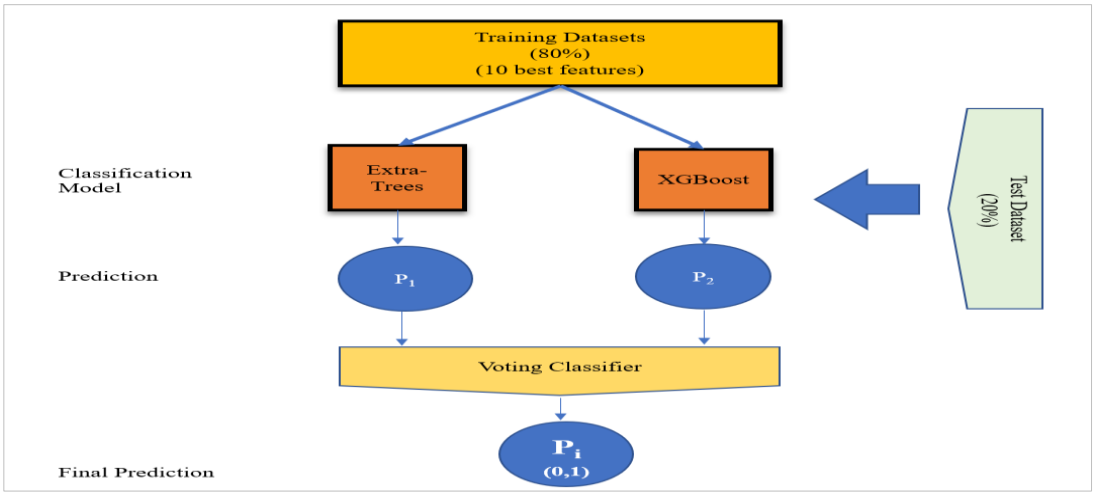

Figure 9 Voting ensemble prediction concept

4.Results

This research experiment was conducted utilizing a 2.80GHz Intel(R) Core (TM) i7-7700HQ processor and 32GB of RAM. The study's dataset source (labeled Ethereum transactions) was obtained from Kaggle. Python and the Jupyter platform are utilized during the analysis and model development processes. The proposed model's performance measurement is evaluated not only from the perspective of accuracy in making decisions. Therefore, the assessment of model performance measurement needs to consider the values of recall, precision, and F-Score. The measurements (accuracy, recall, precision, and F-Score) result from the confusion matrix for the classification of actual values and predicted values (see *Figure 10*), which consists of:

True Positive (TP) – Correctly identified positive values

True Negative (TN) – Correctly identified negative values

False Positive (FP) – Positive values identified as negative (also called Type I Error)

False Negative (FN)- Negative values identified as positive (also called Type II Error)

This section shows the experimental findings by comparing the performance of five individual classifiers and ensemble voting classifiers using accuracy, precision, recall, and the F score (see *Table 7*).

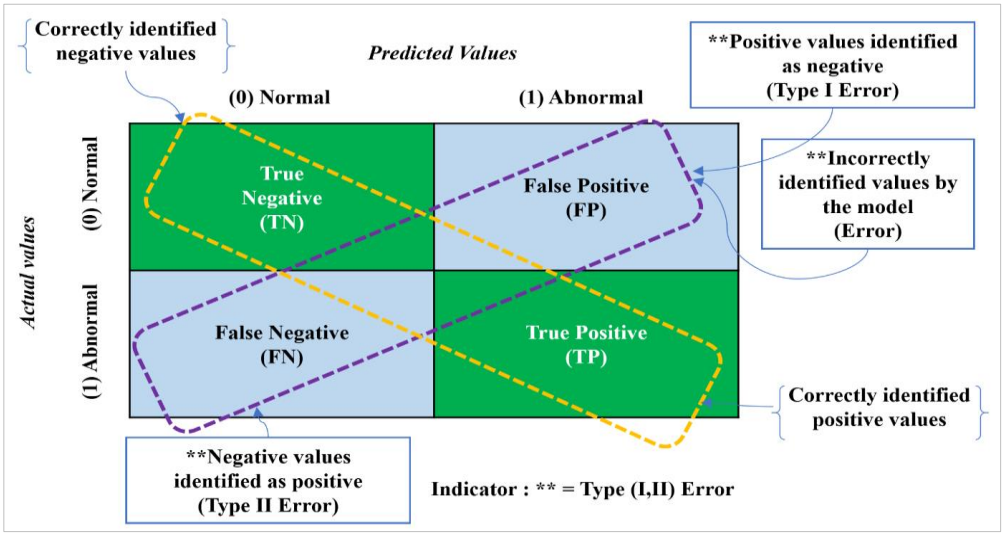

Figure 10 Classification confusion matrix

Table 7 The summary of the performance evaluation matrix

Measurement	Precision (P)	Recall (R)	F1-score	Accuracy	True positive (TP)	False positive (FP)	True negative (TN)	False negative (FN)
Formula	**TP /(TP+FP) (3)**	**TP /(TP+FN) (4)**	**2.P.R /(P+R) (5)**	**(TP+TN) /(TP+TN+FP+FN) (6)**	**TP**	**FP**	**TN**	**FN**
Definition	Displays the amount of accurate positive forecasts	It reports the proportion of positive class forecasts that were accurate for the positive class.	F1 represents the mean of precision and recall.	The proportion of cases in which the model correctly predicted the value.	Actual is abnormal and predicted abnormal class	Actual is normal but predicted abnormal	Actual is normal and predicted normal	Actual abnormal but predicted normal
Description	A high precision number indicates that your model produces few false positives.	A low recall value indicates that your model generates several false negatives.	High score indicative of the good performance	A high value indicative of good performance	A high value indicative of good performance	Lower values indicate performance excellence.	A high value indicative of good performance	Lower values indicate performance excellence.

4.1Evaluation of proposed features

This section examines the experimental results to determine the efficacy of the proposed framework's features. The results of this analysis, presented in *Table 8*, indicate that the proposed set of features is capable of detecting anomalies. In detail, the effectiveness of the proposed study was observed based on a comparison of the accuracy performance values of the model produced from the use of full features (50) and proposed features (10) (See *Figure 11*). The proposed model successfully reduces the dimension size of features from 50 to 10 through the best feature selection method. The results of the analysis show that the accuracy value of the model produced using the proposed study has increased from 95.78% (50 features) to 96.78% (10 features). The effectiveness of this model is also tested by using its full features (50). The results found that the accuracy value has increased from 95.78% (full features without the proposed model) to 98.34% (full features using the proposed model). The classification time (training and testing time) for the proposed model that uses the proposed features (10) is more optimal compared to the full features (50) with the proposed model, based on the decrease in the reading of the classification time (in milliseconds). Therefore, two things can be concluded from this study: First, the performance of the proposed model is better using only the 10 best features compared to the full 50 features. The second observation found that the proposed model is effective for anomaly detection based on its ability to produce higher accuracy values using proposed features and full features.

Table 8 Evaluation of proposed features performance

No	Features	Features size	Accuracy (%)	Training time(ms)	Testing time (ms)
1	Full features without the proposed model	50	95.78	0.126003027	0.00200057
2	Full features with the proposed model	50	98.34	2.007027149	0.101966619
3	**Proposed features with the proposed model**	**10**	**96.78**	**1.359968424**	**0.092002153**

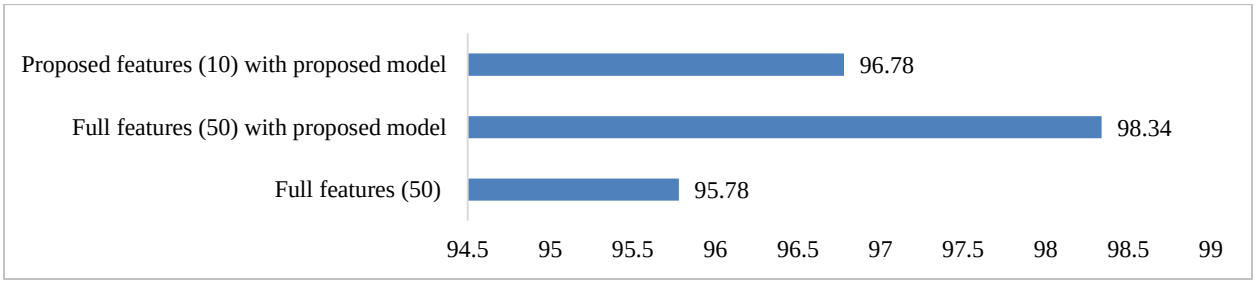

Figure 11 Accuracy of every proposed feature and full features

4.2 Evaluation of proposed ensemble model

The effectiveness of the voting ensemble model (Extra-trees, XGB) is tested by comparing the performance of this model with individual classifier models. The summary of the experimental results is displayed in *Table 9*. The proposed ensemble model successfully produced the highest accuracy value (96.78%) compared to dominant classifiers such as RF (96.04%) and XGB (96.68%). In addition to the aspect of accuracy measurement, this ensemble model performed very well in aspects of precision, F1-Score, recall, true positive rate (TPR), false positive rate (FPR), false negative rate (FNR), and true negative rate (TNR) compared to other classifiers (see *Figure 12*).

Table 9 Comparison of model ensemble performance with various individual classifiers

No	Approach	Precision	F1-Score	Recall	Accuracy	Roc_auc_score (%)	TPR	FPR	FNR	TNR
1	RF	96.89	96.01	95.15	96.04	96.04	0.95	0.03	0.05	0.97
2	KNN	94.84	95.13	95.41	95.11	95.11	0.95	0.05	0.05	0.94
3	DT	91.21	91.48	91.76	91.46	91.46	0.92	0.08	0.08	0.91
4	LR	86.2	87.3	88.43	87.13	87.13	0.88	0.14	0.12	0.86
5	XGB	96.25	96.69	97.14	96.68	96.68	0.97	0.04	0.03	0.96
7	**Proposed Model**	**97.74**	**96.78**	**96.81**	**96.78**	**96.78**	**0.97**	**0.03**	**0.03**	**0.97**

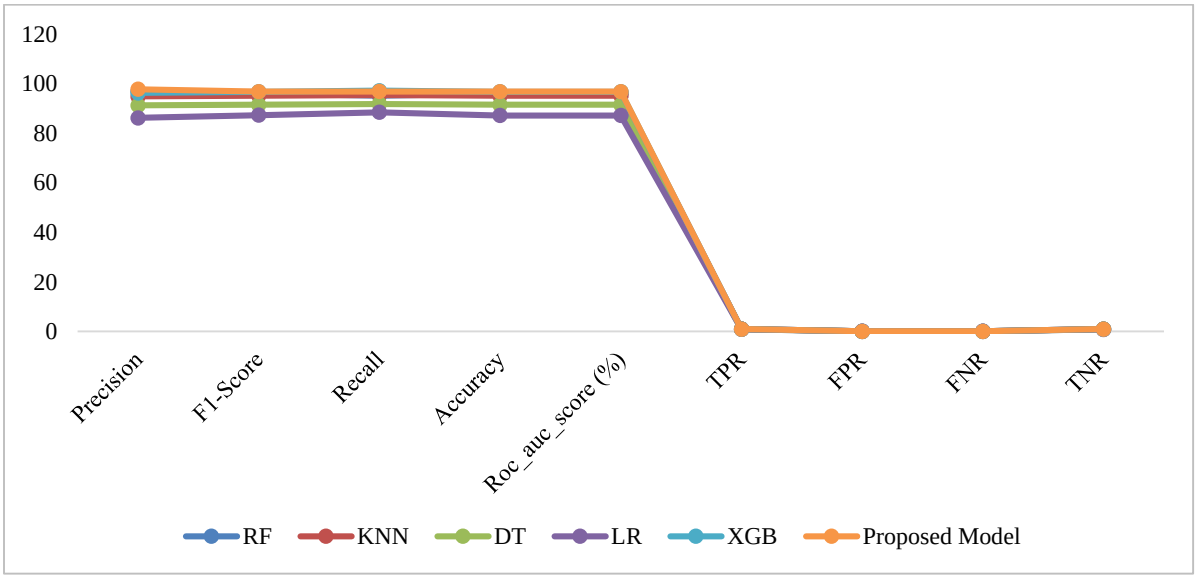

Figure 12 Metric performance for ensemble model testing result with various individual classifiers

Misclassified rate analysis showed that the ensemble model managed to produce the lowest FPR and FNR rate (0.03) compared to other classifiers (low FPR and FNR indicate better model performance), as shown in *Figure 13*.

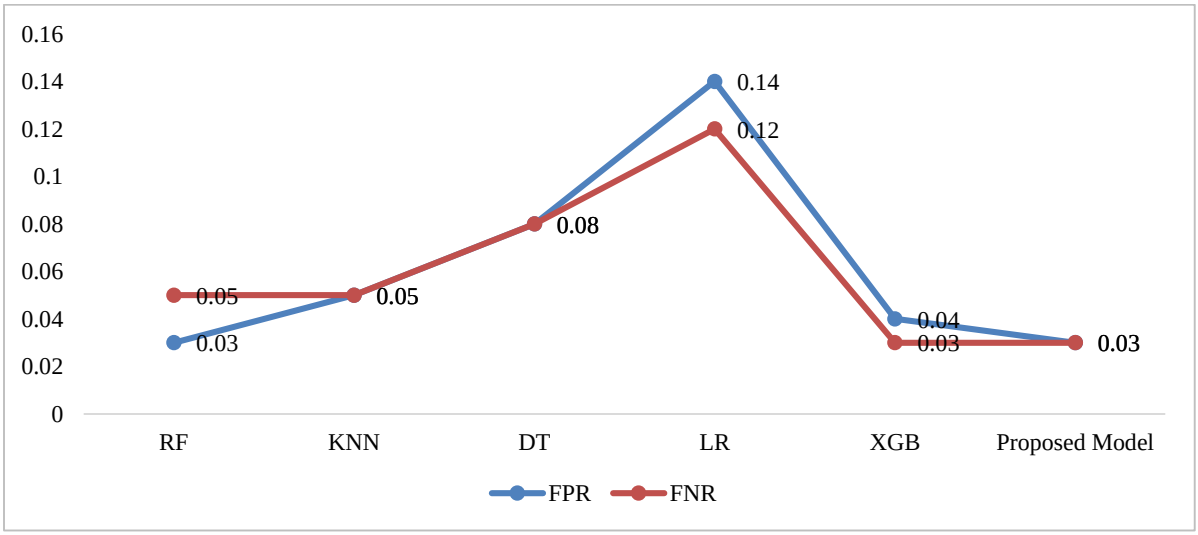

Figure 13 Misclassified rate analysis using FPR and FNR

These results show that the proposed model that uses an ensemble soft voting approach through a combination of Extra-Trees and XGB classifiers shows the best performance among bagging and boosting models. The XGB model (a stand-alone boosting classifier) shows the second-highest performance after the ensemble approach, with an accuracy value of 96.68%. Through this ensemble combination, the weakness of the weak classifier model can be improved to produce a higher-performing model.

The effectiveness of the proposed model is tested by comparing its performance with other research results. Therefore, a comparative analysis of model performance results between these studies has been carried out using the same dataset. The results of the analysis indicate that the proposed model (ensemble voting) has a higher accuracy value (98.70%) than the XGB classifier accuracy value (95.30%) discovered in the previous study, as shown in *Table 10*. The ensemble approach effectively enhances model performance, compensates for the shortcomings of weak classifiers, and reduces variance.

4.3 Comparison with existing works

Table 10 Comparison of the proposed method with previous work

No.	Research	Approach	Dataset size	Best features size	Accuracy (%)
1	[25]	XGB	9841,50	10	95.3
2	**Proposed ensemble model**	**Voting ensemble**	**9841,50**	**10**	**96.78**

5. Discussion

This study employs a data set of size (9841, 50) extracted from the Kaggle public repository containing Ethereum history transactions. Other research work also uses this dataset to analyze Ethereum transaction anomalies. However, the research's outcomes vary based on the proposed study methodology. The selection of features is the most crucial aspect of dataset analysis because it substantially impacts the investigation's efficacy. This study aims to investigate new techniques or methods for generating the best features by employing multiple features filtering techniques, including MI, ANOVA, and RFE. This method is combined with the feature importance technique employing the RF classifier to generate the ten best features in descending order of importance. Even though the number of features has been reduced from 50 to 10, the choice or strategy for employing the classifier also influences the production of the final model. This study investigates the ensemble approach by combining Extra-Trees

(bagging) and XGB (boosting) classifiers via the ensemble soft voting technique.

Experimental results show that the ensemble model that uses the proposed best features successfully produces the highest accuracy value (96.78%) compared to other classifiers. The XGB classifier model is the second highest (96.68%), and the RF is in the third position (96.04%). The effectiveness of this proposed model is tested by comparing the accuracy values produced with those of the model that uses full features (50). The results of this comparison clearly show that the proposed model successfully increased the accuracy value from 95.78% to 98.34%. In terms of time classification rate (in milliseconds), the proposed model with the best features (10) exhibits an optimal classification time (lower) compared to the proposed model that uses full features (50). The performance of the model is also measured from the perspective of misclassified rates (FPR and FNR). The low values on FPR and FNR show that the performance of the model is better. The results of this analysis show that the proposed ensemble model successfully produces the lowest FPR and FNR rates (0.03) compared to individual classifiers. This study also managed to produce better accuracy values (96.78%) compared to other studies (95.3%) that used the same dataset. Overall, the ensemble voting model that uses the 10 best features successfully produces a higher-performance anomaly detection model.

However, this study has limitations in terms of the blockchain ecosystem's features. This study only analyses the conduct of blockchain transactions. Nonetheless, there are several additional features, such as analysis of the smart contract's behaviour (semantics) via analysis of certain features of the source code, such as OPCODE and application binary interface (ABI) code, and historical transactions associated with the smart contract. Because fraud or malfeasance on the blockchain network can occur in multiple ways, this is the case. Therefore, it is necessary to analyse all potential features in order to increase the effectiveness of anomaly detection.

A complete list of abbreviations is shown in *Appendix I.*

6. Conclusion and future work

The development of Dapp applications in most industrial domains across the globe have presented security challenges. Problems with phishing, smart contract vulnerabilities (bugs and defects), Ponzi schemes, etc., have resulted in losses for people all over the globe. Therefore, it is necessary to implement early detection measures for suspicious activities on the blockchain network to avoid intrusions. If the analysis is performed manually by examining the greatest blockchain transactions one by one, it presents a formidable obstacle. Therefore, blockchain must be combined with another technology, namely ML, in order to develop an anomaly detection model based on the analysis of blockchain's historical transactions. The largest obstacle is that the blockchain contains too much data (high dimensions of features and instances), making data analysis challenging. Consequently, the method of feature selection is employed to ensure that only the most essential features are incorporated into the development of the final model. Efforts to reduce the number of features can improve the model's performance and save money and time.

This study employs three feature filtering techniques (MI, ANOVA, and RFE) and selects the most accurate filtering technique. ANOVA has been selected as the optimal filtering technique, and the ten best features have been chosen using the feature importance technique and the RF classifier. The dataset is then trained and evaluated using multiple individual classifier models and an ensemble-based recommendation model. The performance of the final model was evaluated using a soft voting ensemble comprised of two different combinations of bagging (Extra-trees) and boosting (XGB) classifiers. The results of the study indicate that ensemble voting generates high accuracy (96.78%) as well as high precision, recall, and F1-Score values. The accuracy value of the proposed model is also superior to the accuracy (95.3%) generated by the previous study's performance. This study also concluded that the size of features was effectively reduced from fifty to ten, resulting in increased performance and decreased classification time.

Blockchain data analysis is challenging because it requires high-spec machines, including central processing unit (CPU), random access memory (RAM), and storage. Processing using a ML approach requires sufficient memory if the size of the data dimension is too large, especially when involving the analysis of hybrid features (a combination of contract features, transaction behavior, and source code). The analysis of anomalies in the blockchain network does not only focus on historical transactions. This is because the attackers use various methods to carry out criminal activities by finding loopholes in the blockchain. Therefore, among the potential studies in the future is the analysis of smart contract source code

(semantics) for anomaly detection. Smart contracts are source codes developed using the Solidity language to control business case logic. Therefore, this smart contract is exposed to low code quality (bugs or defects) to the point that hackers manipulate this source code for illegal money transfers. Future research also could concentrate on enhancing model performance by performing the engineering procedures of feature selection and hyper-parameter optimization on each model using the most suitable methods. In conclusion, the study for anomaly analysis on historical Ethereum transactions succeeded in producing a more accurate level of anomaly detection through the feature filter method, feature extraction, and ensemble learning approaches.

Acknowledgment
None.

Conflicts of interest
The authors have no conflicts of interest to declare.

Author's contribution statement
Sabri Hisham: Models and method selection, building a framework, conducting experiments, analysis of experimental results, draft writing, checking for plagiarism, and proofreading**. Mokhairi Makhtar:** Supervision, gave an opinion, input on draught revision, and final revision. **Azwa Abdul Aziz:** Supervision, exchange of sample manuscripts, and draught evaluation comments

References
[1] Jatoth C, Jain R, Fiore U, Chatharasupalli S. Improved classification of blockchain transactions using feature engineering and ensemble learning. Future Internet. 2021; 14(1):1-12.
[2] Hou H. The application of blockchain technology in E-government in China. In 26th international conference on computer communication and networks 2017 (pp. 1-4). IEEE.
[3] Roehrs A, Da CCA, Da R. OmniPHR: a distributed architecture model to integrate personal health records. Journal of Biomedical Informatics. 2017; 71:70-81.
[4] Sidhu J. Syscoin: a peer-to-peer electronic cash system with blockchain-based services for e-business. In 26th international conference on computer communication and networks 2017 (pp. 1-6). IEEE.
[5] Hakak S, Khan WZ, Gilkar GA, Imran M, Guizani N. Securing smart cities through blockchain technology: architecture, requirements, and challenges. IEEE Network. 2020; 34(1):8-14.
[6] Thompson S. The preservation of digital signatures on the blockchain. See Also. 2017; 31(3):1-17.
[7] Togawa Y. Nomure research institute: survey on blockchain technologies and related services. Information Economy Division Commerce and Information Policy Bureau. 2016.
[8] Hisham S, Makhtar M, Aziz AA. A comprehensive review of significant learning for anomalous transaction detection using a machine learning method in a decentralized blockchain network. International Journal of Advanced Technology and Engineering Exploration. 2022; 9(95):1366-96.
[9] Kanan T, Obaidat AT, Al-lahham M. SmartCert blockchain imperative for educational certificates. In Jordan international joint conference on electrical engineering and information technology 2019 (pp. 629-33). IEEE.
[10] Moosavi J, Naeni LM, Fathollahi-fard AM, Fiore U. Blockchain in supply chain management: a review, bibliometric, and network analysis. Environmental Science and Pollution Research. 2021:1-5.
[11] Zheng Z, Xie S, Dai HN, Chen X, Wang H. Blockchain challenges and opportunities: a survey. International Journal of Web and Grid Services. 2018; 14(4):352-75.
[12] Bahga A, Madisetti VK. Blockchain platform for industrial internet of things. Journal of Software Engineering and Applications. 2016; 9(10):533-46.
[13] Eduardo A, Sousa J, Oliveira VC, Almeida VJ, Borges VA, Bernardino HS, et al. Fighting under-price DoS attack in ethereum with machine learning techniques. ACM SIGMETRICS Performance Evaluation Review. 2021; 48(4):24-7.
[14] Chen W, Zheng Z, Cui J, Ngai E, Zheng P, Zhou Y. Detecting Ponzi schemes on Ethereum: towards healthier blockchain technology. In proceedings of the world wide web conference 2018 (pp. 1409-18).
[15] Meiklejohn S, Pomarole M, Jordan G, Levchenko K, Mccoy D, Voelker GM, et al. A fistful of bitcoins: characterizing payments among men with no names. In proceedings of the conference on internet measurement conference 2013 (pp. 127-40). ACM.
[16] Khonji M, Iraqi Y, Jones A. Phishing detection: a literature survey. IEEE Communications Surveys & Tutorials. 2013; 15(4):2091-121.
[17] Agarwal R, Barve S, Shukla SK. Detecting malicious accounts in permissionless blockchains using temporal graph properties. Applied Network Science. 2021; 6(1):1-30.
[18] Wen H, Fang J, Wu J, Zheng Z. Transaction-based hidden strategies against general phishing detection framework on ethereum. In international symposium on circuits and systems 2021 (pp. 1-5). IEEE.
[19] Jung E, Le TM, Gehani A, Ge Y. Data mining-based ethereum fraud detection. In 2019 international conference on blockchain (Blockchain) 2019 (pp. 266-73). IEEE.
[20] Preuveneers D, Rimmer V, Tsingenopoulos I, Spooren J, Joosen W, Ilie-zudor E. Chained anomaly detection models for federated learning: an intrusion detection case study. Applied Sciences. 2018; 8(12):1-21.
[21] Pham T, Lee S. Anomaly detection in bitcoin network using unsupervised learning methods. arXiv preprint arXiv:1611.03941. 2016; 1611:03941.
[22] Yan Z, Susilo W, Bertino E, Zhang J, Yang LT. AI-driven data security and privacy. Journal of Network and Computer Applications. 2020; 172:102842.

[23] Hisham S, Makhtar M, Aziz AA. Combining multiple classifiers using ensemble method for anomaly detection in blockchain networks: a comprehensive review. International Journal of Advanced Computer Science and Applications. 2022; 13(8):404-22.
[24] Aljofey A, Rasool A, Jiang Q, Qu Q. A feature-based robust method for abnormal contracts detection in ethereum blockchain. Electronics. 2022; 11(18):1-24.
[25] Farrugia S, Ellul J, Azzopardi G. Detection of illicit accounts over the Ethereum blockchain. Expert Systems with Applications. 2020; 150:113318.
[26] Sallam A, Rassem T, Abdu H, Abdulkareem H, Saif N, Abdullah S. Fraudulent account detection in the Ethereum's network using various machine learning techniques. International Journal of Software Engineering and Computer Systems. 2022; 8(2):43-50.
[27] Ibrahim RF, Elian AM, Ababneh M. Illicit account detection in the Ethereum blockchain using machine learning. In international conference on information technology 2021 (pp. 488-93). IEEE.
[28] Baba NM, Makhtar M, Fadzli SA, Awang MK. Current issues in ensemble methods and its applications. Journal of Theoretical & Applied Information Technology. 2015; 81(2):266-76.
[29] Bulusu S, Kailkhura B, Li B, Varshney PK, Song D. Anomalous example detection in deep learning: a survey. IEEE Access. 2020; 8:132330-47.
[30] Zhang YL, Li L, Zhou J, Li X, Zhou ZH. Anomaly detection with partially observed anomalies. In companion proceedings of the web conference 2018 (pp. 639-46). ACM.
[31] Signorini M, Pontecorvi M, Kanoun W, Di PR. BAD: a blockchain anomaly detection solution. IEEE Access. 2020; 8:173481-90.
[32] Mirsky Y, Golomb T, Elovici Y. Lightweight collaborative anomaly detection for the IoT using blockchain. Journal of Parallel and Distributed Computing. 2020; 145:75-97.
[33] Chen W, Zheng Z, Ngai EC, Zheng P, Zhou Y. Exploiting blockchain data to detect smart ponzi schemes on ethereum. IEEE Access. 2019; 7:37575-86.
[34] Hu T, Liu X, Chen T, Zhang X, Huang X, Niu W, et al. Transaction-based classification and detection approach for Ethereum smart contract. Information Processing & Management. 2021; 58(2):102462.
[35] Sosu RN, Chen J, Brown-acquaye W, Owusu E, Boahen E. A vulnerability detection approach for automated smart contract using enhanced machine learning techniques. Europe PMC. 2022; 1-10.
[36] Han D, Li Q, Zhang L, Xu T. A smart contract vulnerability detection model based on syntactic and semantic fusion learning. Wireless Communications and Mobile Computing. 2023; 2023:1-12.
[37] Ashizawa N, Yanai N, Cruz JP, Okamura S. Eth2Vec: learning contract-wide code representations for vulnerability detection on Ethereum smart contracts. In proceedings of the 3rd international symposium on blockchain and secure critical infrastructure 2021 (pp. 47-59). ACM.
[38] Huang J, Zhou K, Xiong A, Li D. Smart contract vulnerability detection model based on multi-task learning. Sensors. 2022; 22(5):1-24.
[39] Huang Y, Kong Q, Jia N, Chen X, Zheng Z. Recommending differentiated code to support smart contract update. In 27th international conference on program comprehension 2019 (pp. 260-70). IEEE.
[40] Wu J, Yuan Q, Lin D, You W, Chen W, Chen C, et al. Who are the phishers? phishing scam detection on ethereum via network embedding. IEEE Transactions on Systems, Man, and Cybernetics: Systems. 2020; 52(2):1156-66.
[41] Jin C, Jin J, Zhou J, Wu J, Xuan Q. Heterogeneous feature augmentation for ponzi detection in ethereum. IEEE Transactions on Circuits and Systems II: Express Briefs. 2022; 69(9):3919-23.
[42] Nerurkar P, Busnel Y, Ludinard R, Shah K, Bhirud S, Patel D. Detecting illicit entities in bitcoin using supervised learning of ensemble decision trees. In proceedings of the 10th international conference on information communication and management 2020 (pp. 25-30). ACM.
[43] Yang X, Chen Y, Qian X, Li T, Lv X. BCEAD: a blockchain-empowered ensemble anomaly detection for wireless sensor network via isolation forest. Security and Communication Networks. 2021; 2021:1-10.
[44] Nerurkar P, Bhirud S, Patel D, Ludinard R, Busnel Y, Kumari S. Supervised learning model for identifying illegal activities in bitcoin. Applied Intelligence. 2021; 51:3824-43.
[45] Bhowmik M, Chandana TS, Rudra B. Comparative study of machine learning algorithms for fraud detection in blockchain. In 5th international conference on computing methodologies and communication 2021 (pp. 539-41). IEEE.
[46] Poursafaei F, Hamad GB, Zilic Z. Detecting malicious ethereum entities via application of machine learning classification. In 2nd conference on blockchain research & applications for innovative networks and services 2020 (pp. 120-7). IEEE.
[47] Fan S, Fu S, Xu H, Zhu C. Expose your mask: smart ponzi schemes detection on blockchain. In international joint conference on neural networks 2020 (pp. 1-7). IEEE.
[48] Yuan Q, Huang B, Zhang J, Wu J, Zhang H, Zhang X. Detecting phishing scams on ethereum based on transaction records. In international symposium on circuits and systems 2020 (pp. 1-5). IEEE.
[49] Chen J, Xia X, Lo D, Grundy J, Luo X, Chen T. Defectchecker: automated smart contract defect detection by analyzing EVM bytecode. IEEE Transactions on Software Engineering. 2021; 48(7):2189-207.
[50] Buterin V. A next-generation smart contract and decentralized application platform. Ethereum White Paper. 2014:1-36.

Sabri Hisham completed his Bachelor's degree in computer science (Industrial Computing) from Universiti Teknologi Malaysia (UTM) in 2001. He obtained his Master's degree in software engineering from Universiti Malaysia Pahang (UMP) in 2014. Currently, he is a Ph.D. student in the Department of Computer Science at Universiti Sultan Zainal Abidin, located in the Faculty of Computing and Informatics in Terengganu, Malaysia. Additionally, Sabri Hisham holds the position of Head of Infostructure at the Information and Technology Department of Universiti Malaysia Pahang. He is recognized as an expert and certified professional in Blockchain Solidity Smart Contracts and Ethereum. His current research interests revolve around areas such as Blockchain, Bitcoin, Machine Learning (ML), Internet of Things (IoT), Mobile Apps, Web Applications, SCADA, and Telemetry Systems.
Email: sabrihisham@ump.edu.my

Mokhairi Makhtar completed his Ph.D. in 2012 at the University of Bradford in the United Kingdom. Presently, he holds the position of Professor in the Department of Computer Science at Universiti Sultan Zainal Abidin, located in Terengganu, Malaysia. His research interests encompass various areas, including Machine Learning, Ensemble Methods, Data Mining, Soft Computing, Timetabling and Optimization, Natural Language Processing, E-Learning, and Deep Learning.
Email: mokhairi@unisza.edu.my

Azwa Abdul Aziz received a Bachelor's degree in computer science from Universiti Teknologi Mara, Malaysia, in 2002. He further pursued his studies and completed a Master's degree in computer science from the University of Malaysia Terengganu in 2010. Currently, he serves as a lecturer in the Department of Computer Science at Sultan Zainal Abidin University in Terengganu, Malaysia. Azwa Abdul Aziz's research interests encompass various fields, including Big Data Analytics, Text Mining, Business Intelligence, and Machine Learning.
Email: azwaaziz@ unisza.edu.my

Appendix I

S. No.	Abbreviations	Descriptions
1	ABI	Application Binary Interface
2	AdaBoost	Adaptive Boosting
3	AI	Artificial Intelligence
4	ANOVA	Analysis of Variance
5	BCEAD	Blockchain-Empowered Ensemble Anomaly Detection
6	CA	Central Authority
7	CFG	Control Flow Graph
8	CNN	Convolutional Neural Network
9	CPU	Central Processing Unit
10	Dapp	Decentralized Application
11	DT	Decision Tree
12	DeFi	Decentralized Financial
13	DL	Deep Learning
14	DNN	Deep Neural Network
15	FN	False Negative
16	FNR	False Negative Rate
17	FP	False Positive
18	FPR	False Positive Rate
19	GSA	Gravitational Search Algorithms
20	HFAug	Heterogeneous Feature Augmentation
21	IDEs	Integrated Development Environments
22	IF	Isolation Forest
23	IoT	Internet of Things
24	IP	Internet Protocol
25	KNN	K-Nearest Neighbor
26	LR	Logistic Regression
27	LSTM	Long Short-Term Memory
28	MI	Mutual Information
29	ML	Machine Learning
30	MLP	Multilayer Perceptron
31	NB	Naïve Bayes
32	NFT	Non-Fungible Tokens
33	NLP	Natural Language Processing
34	OPCODE	Operation Code
35	P2P	Peer-to-peer
36	PSO	Particle Swarm Optimization
37	RAM	Random Access Memory
38	RF	Random Forest
39	RFE	Recursive Feature Elimination
40	SMOTE	Synthetic Minority Oversampling Technique
41	SVM	Support Vector Machine
42	TN	True Negative
43	TNR	True Negative Rate
44	TP	True Positive
45	TPR	True Positive Rate
46	TSVM	Transductive Support Vector Machines
47	WSN	Wireless Sensor Networks
48	XGB	Extreme Gradient Boosting