Karbala International Journal of Modern Science

Manuscript 3229

A Machine Learning based Approach to Detect the Ethereum Fraud Transactions with Limited Attributes

Rabia Musheer Aziz

Mohammed Farhan Baluch

Sarthak Patel

Pavan Kumar

Follow this and additional works at: https://kijoms.uokerbala.edu.iq/home

Part of the Biology Commons, Chemistry Commons, Computer Sciences Commons, and the Physics Commons

A Machine Learning based Approach to Detect the Ethereum Fraud Transactions with Limited Attributes

Abstract

Ethereum smart contracts have recently received new commercial applications and a lot of attention from the scientific community. Ethereum eliminates the requirement for a trusted third party by allowing untrusted parties to expose contract details in computer code. Nonetheless, as online commerce grows, plenty of fraudulent activities, such as money laundering, bribery, and phishing, emerge as major threats to trade security. For correctly recognizing fraudulent transactions, this paper developed a Light Gradient Boosting Machine (LGBM) technique-based model. The modified LGBM model optimized the parameters of Light GBM using the Euclidean distant structured estimation approach. This paper also examines the performance of different popular models such as Random Forest (RF), Multi-Layer Perceptron (MLP), Logistic Regression, k-Nearest Neighbors (KNN), XGBoost, Support Vector Classification (SVC), and ADAboost with limited features and compares their performance metrics with the proposed model for Ethereum fraudulent activity classification. A comparative performance evaluation matrices scores of different popular models along with the proposed model demonstrated the applicability of the proposed approach. The modified LGBM algorithms and RF models demonstrate the best performance compared to other models with the highest accuracies, while the modified LGBM algorithm has a slightly superior performance of 99.17 percent compared to the RF model's 98.26 percent.

Keywords

k-Nearest Neighbors (KNN); Random Forest (RF); Support Vector Classification (SVC); Ethereum Fraud; Logistic Regression.

Creative Commons License

This work is licensed under a Creative Commons Attribution-Noncommercial-No Derivative Works 4.0 License.

RESEARCH PAPER

A Machine Learning Based Approach to Detect the Ethereum Fraud Transactions with Limited Attributes

Rabia Musheer Aziz, Mohammed Farhan Baluch, Sarthak Patel, Pavan Kumar*

School of Advanced Sciences & Languages, VIT Bhopal University, Bhopal-Indore Highway, Kothrikalan, Sehore, MP-466114, India

Abstract

Ethereum smart contracts have recently received new commercial applications and a lot of attention from the scientific community. Ethereum eliminates the requirement for a trusted third party by allowing untrusted parties to expose contract details in computer code. Nonetheless, as online commerce grows, plenty of fraudulent activities, such as money laundering, bribery, and phishing, emerge as major threats to trade security. For correctly recognizing fraudulent transactions, this paper developed a Light Gradient Boosting Machine (LGBM) technique-based model. The modified LGBM model optimized the parameters of Light GBM using the Euclidean distant structured estimation approach. This paper also examines the performance of different popular models such as Random Forest (RF), Multi-Layer Perceptron (MLP), Logistic Regression, k-Nearest Neighbors (KNN), XGBoost, Support Vector Classification (SVC), and ADAboost with limited features and compares their performance metrics with the proposed model for Ethereum fraudulent activity classification. A comparative performance evaluation matrices scores of different popular models along with the proposed model demonstrated the applicability of the proposed approach. The modified LGBM algorithms and RF models demonstrate the best performance compared to other models with the highest accuracies, while the modified LGBM algorithm has a slightly superior performance of 99.17 percent compared to the RF model's 98.26 percent.

Keywords: k-Nearest Neighbors (KNN), Random forest (RF), Support Vector Classification (SVC), Ethereum fraud, Logistic regression

1. Introduction

Ethereum is a famous cryptocurrency exchange platform as well as the most well-known platform for peer-to-peer programming. Blockchain security and supervision have recently received a lot of attention [1,2]. Ethereum is an open-source blockchain technology that allows smart contracts to be implemented. Its emergence solves the problem of the Bitcoin protocol's limited scalability. Developers can construct general-purpose smart contracts using the Solidity language because of its adaptability [3,4]. Some fraudulent smart contracts may exist among the many smart contracts, stealing ether from network participants. Because of its simplicity and adaptability, Ethereum has grown swiftly. Ethereum has also surpassed Bitcoin as the second most valuable cryptocurrency. Since then, Ethereum and Bitcoin have grown in popularity, particularly among large corporations, because they were invented by Clohessy et al. [1], Li and Whinston [2], as well as Liu and Serletis [3]. The main goal is to give power to the individual, letting employers control their data and transactions. Ethereum, created by Vitalik Buterin, is one such cryptocurrency. Ethereum, for those who are new, is a cryptocurrency allocation mechanism that allows you to transmit cryptocurrency to anybody for a little fee. As discussed by Leal et al. [5], and Panarello et al. [6], as Ethereum is built on a blockchain network, anyone can take advantage of the service of digital transactions with very minor transactional charges in a safe way. According to Zhao and Liu [7], Kabainskas and Šutienė [8], as well as Brauneis et al.

Received 3 September 2021; revised 29 March 2022; accepted 30 March 2022.
Available online 1 May 2022

* Corresponding author at:
E-mail addresses: rabia.aziz2010@gmail.com (R.M. Aziz), farhanbaluch1301@gmail.com (M.F. Baluch), sarthak3136@gmail.com (S. Patel), pavankmaths@gmail.com (P. Kumar).

https://doi.org/10.33640/2405-609X.3229
2405-609X/© 2022 University of Kerbala. This is an open access article under the CC-BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

[9], a blockchain is a disseminated, deconcentrated public bank account that authenticates and records every transaction. As a result, it is quite slow but very secure.

Cryptocurrencies are a type of decentralized currency, which means that a third-party person cannot seize control of the blockchain network. It would be difficult to identify users who were responsible for the scam if it occurred. These considerations, as well as the users' anonymity, might lead to fraudulent behaviour. The Ponzi scheme is one of the most well-known Ethereum scams [9].

Cryptocurrencies are decentralized, which means that no one body can acquire control of the blockchain. It would be difficult to identify users who were responsible for the scam if it occurred. These considerations, as well as the users' anonymity, might lead to fraudulent behaviour. The Ponzi scheme is one of the most well-known Ethereum scams. These schemes masquerading as secure investment schemes have proliferated on Ethereum, as proposed by Jung et al. [10]. As discussed by Bartoletti et al. [11] and Chen et al. [12], Ponzi schemes are structured in such a deceptive way that, in 95% of the cases, early investors in the scheme take most of the money, leaving other investors with a very marginal profit or extreme loss. There are very high chances that the investor's money will get squandered.

A Ponzi scheme can be considered with the topology of the pyramid. At the top of the pyramid is the investor who invested early, and all the following layers compensate for the upper-tier investors. Simply, the lower-level investor in the pyramid pays the one at the top level. The scheme will ultimately fail since it will be difficult to attract new investors. As an outcome, those at the topmost of the pyramid will benefit. Whereas bottom layer investors will get trapped in this scheme and will have very little to no money in hand, as investigated by Bartoletti et al. [13] as well as Vasek and Moore [14].

Taking advantage of the anonymity, crooks rapidly took advantage of the site's status to rake off other users by inventing a slew of scams. Unlike previous frauds, cryptocurrencies are not yet governed by any official standards, making it harder to pursue restitution for losses incurred by fraud. Because of the blockchain's immutability and users' anonymity, it is almost impossible to reverse a fraudulent transaction. It would be difficult and time-consuming to manually search through all of these transactions, looking for any transactions that were deemed to have abnormal features. The rapid formation of such blocks on the network, notably transactions and smart contracts, suggests the adoption of machine learning (ML) techniques to help in the detection of any trends linked with aberrant activity. Several studies on identifying Ethereum scams are being conducted using various classifiers and methodologies. Some of these are detailed below:

Ajay et al. [15] demonstrated a technique for identifying abnormalities in Ethereum networks and determining whether users are suspicious using machine learning algorithms. He employed decision trees and random forests, among other machine learning techniques. When indegree nodes are taken into account, the algorithms' accuracy is 83.66 percent and 98.93 percent, respectively.

Teng et al. [16] used a pattern-based approach to identify the anomalous Ethereum smart contracts. They trained the model using LSTM, that is, long-term short-term memory, as well as by data slicing. The findings demonstrated a high level of precision in contract identification.

Runnan et al. [17] suggested a way to detect ambiguous transactions with the help of a graph convolutional network model. They used a single model to detect the transactions and achieved an accuracy of 95% with the help of neural networks. They used web crawlers to capture the addresses of fraud transactions.

Yuan et al. [18] created a machine learning model for detecting phishing scams on the Ethereum market. Using the transaction information, they first established the transaction network model to predict the fraudulent transaction. The next model was trained using the node2vec embedding technique and the SVM classifier. This modified model's final accuracy was 84.6 percent.

Rahmeh et al. [19] put forward a mechanism to detect illicit transactions with the help of supervised learning as well as ensemble learning methods like random forest and decision trees to achieve their results. They used the correlation coefficient to build a dataset with six features on which they trained the model and predicted the accuracy.

The approach described in [20] is based on account attributes, and it also includes opcode functionalities based on the contract's code that is recorded on the blockchain. They constructed three classification models with XGBoost: Opcode, Account, and Account plus Op-code. The third model, i.e., the Account plus Opcode model, outperformed the others with a precision of 94 percent and a recall of 81 percent.

To detect Ponzi schemes structured as smart contracts on the Ethereum network, Chen et al. [21] used supervised learning methods to classify accounts depending on their executable program

(Opcode). Even with experimentally generated contracts, the enhanced Random Forest identified 305 of the 394 smart-Ponzi schemes with probabilistic confidence of more than 90% after parameter modifications.

The motive of this research is to improve the proposed model, which can capture fraudulent activities on the blockchain. Anomalies in the transactional data of the Ethereum blockchain are checked. Abnormal or suspicious transactions are those that deviate from the usual. Furthermore, these transactions may be legal or illegal, but they are worth investigating. We conducted an extensive experiment for a thorough comparison of different machine learning models using various performance criteria.

1.1. Paper objectives

- The Light GBM model has been implemented with Euclidean distance measures to predict fraudulent transactions with faster efficiency and low memory usage.
- The compatibility of gradient boosting with large datasets has been taken into account, considering that it can keep up with the growth of the Ethereum network dataset and still predict accurately.
- To propose a machine learning model which can find fraud activities on the Ethereum blockchain.
- To offer a machine learning strategy for detecting Ethereum fraud with a small number of features.
- In-depth comparison and analysis of several classifiers suggested using various performance metrics.

1.2. Proposed novel work

From the literature review, it was observed that most previous work for detecting fraudulent transactions involved machine learning techniques and optimization techniques with, Random Forest, Decision Tree, Support Vector Machine (SVM), etc. However, very little work has been done with the gradient boosting approach, which holds the potential to give results very quickly and accurately. Also, it has been observed that the proposed modified LGBM performed better than several high-performing algorithms for solving problems in different domains compared to Random Forest and other popular techniques. Therefore, the suggested study employs an enhanced machine learning strategy that employs improved gradient enhancement algorithms which has a high degree of accuracy in detecting fraudulent transactions while avoiding overfitting. Figure 1 shows the flow chart of the proposed approach.

1.3. Paper organization

The rest of the paper is organized as follows. Section 2 explores classification models and the fundamentals of each model. In Section 3, data preprocessing and experimental setup are explained. The efficiency metrics of experimental data with different algorithms are included in Section 4. Section 5 closes with a recap of previous work and some ideas for future improvements.

2. Used classifiers

For the sake of fair comparison, this study used the below-defined classifiers for Ethereum fraud transactions with the same attributes.

2.1. Logistic regression

The logistic regression method predicts a binary outcome from a dataset's numerous independent features. It is used to determine whether or not a certain binary event, such as a 0 or 1, will occur. It may be used by sales firms, for example, to persuade consumers to buy or not buy their goods. It is a linear

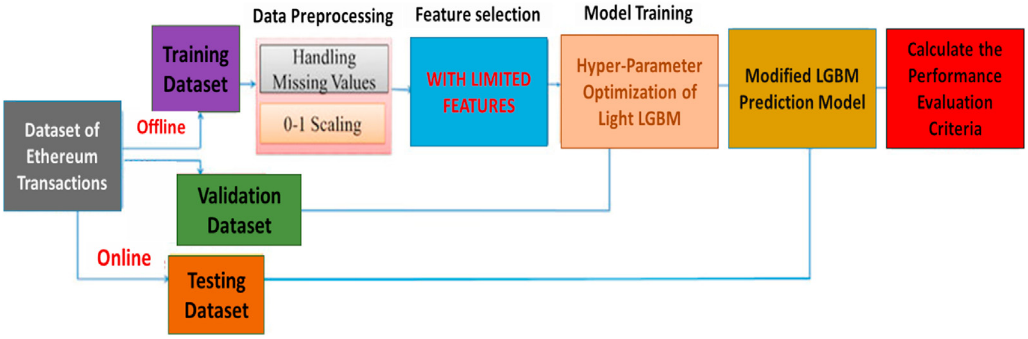

Fig. 1. Flow chart of the proposed modified LGBM based model.

regression model that has been extended. In classification techniques such as logistic regression, we categorize the attribute to be decided or dependent as 0 or 1, but in linear regression, instead of considering 0 and 1 as categories, consider them as values [22]. The diagram shows that the function of logistic regression compresses the linear model graph in the range [0, 1], as discussed by Chen et al. [23] and Arimura et al. [24], as seen in Fig. 2.

2.2. Random forest

The Random Forests (RF) technique is a collaborative learning approach that may be used to conduct classification or other tasks on a dataset by constructing a large number of trees according to the nature of the dataset through training and accordingly predicting the outcome. The below class and probability formula is used to compute the Gini index of every branch on a node, which identifies the most probable branches.

$$Gini = 1 - \sum_{i=1}^{c} (P_i)^2, \tag{1}$$

where *Pi* denotes the type's relative frequency and *c* is the total classes of the dataset. Entropy is determined using a below logarithmic equation, it necessitates more arithmetic than the Gini index. (Sreejith et al. [25] as well as Aziz et al. [26]):

$$Entropy = \sum_{i=1}^{c} P_i * \log_2(P_i) \tag{2}$$

2.3. MLP classifier

A machine learning classification approach is known as a multi-layer perceptron (MLP) classifier. In layman's terms without making things complex, it can be said that it is an artificial neural network (ANN) that is used for classification and implemented with the help of the Scikit-Learn library. This whole network consists of an input layer where inputs for classification are specified, which are multiplied by a self-adjustable weight randomly selected by the machine. Every node in the layers is fully connected to every single node in the subsequent layer. Hidden layers are layers that exist between the input and output layers and may be used as needed by the model. The activation function is applied to the results of multiplication with weights and the results with minimum loss are calculated. Till then, weights get updated iteratively with the help of backpropagation as studied by various researchers [30–32] as well as Desai [33].

2.4. KNN

The clustering algorithm, the K-Nearest Neighbor (KNN) algorithm, is widely used in the data science industry to categorize datasets. It uses the Euclidean distance algorithm to decide the category for the datapoint. Consider a situation in which we have to allocate the datapoint. This can be solved with the help of the KNN as follows:

(a) Pick the K closest neighbours at random.
(b) Determine the nearest neighbour of the datapoint point that needs to be classified using the Euclidean distance.
(c) Count the number of data points in each of these K groups.
(d) Assign the to-be-classified data point to the cluster or neighbours with the greatest number of neighbors.

2.5. XGB

XGBoost is a machine learning approach that enhances the gradient with the help of tree-based

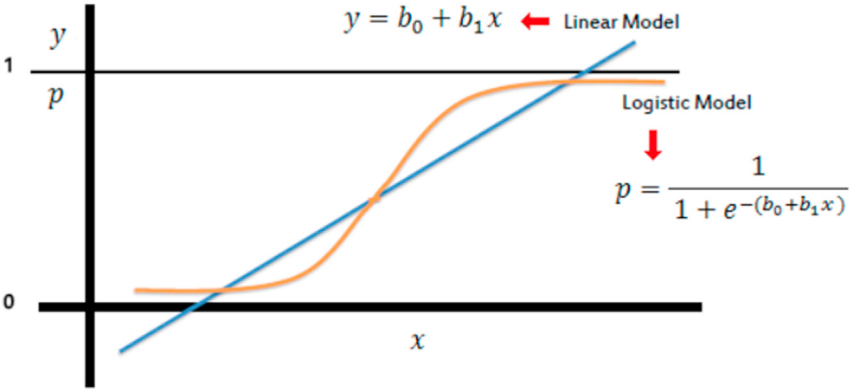

Fig. 2. Graphical Comparison of logistic and linear models.

boosting algorithms to improve the accuracy of classification and regression models. It uses an ensemble method for gradient boosting and supports three different forms of gradient boosting: a stochastic gradient boosting, regularized gradient boosting, and normal boosting. In terms of speed, XGB excels over other gradient boosting approaches. This approach contains an auto pruning function that inhibits the decision tree from growing beyond a particular point, thereby decreasing the overfitting of the machine learning model on our training dataset. It is more precise compared to other similar algorithms.

Boosting algorithms by not compromising memory usage. It is among the most extensively used gradient boosting algorithms due to its numerous benefits as proposed by Farrugia et al. [17].

2.6. SVC

Support vector machines, or SVCs, use a very simple approach to classify the data in a dataset. It distributes the data points by the hyperplane and is very promising in classification and multiple dimensional or non-linear spaces. They provide good results by balancing the memory and provide somewhat faster results with less consumption of resources. The SVC algorithm, according to the inputs, decides on the hyperplane to divide the dataset. As there can be infinite ways to decide on which plane to choose, it is decided by finding the points which are nearest to the plane (support vectors) and computing the distance between all the support vectors and the respective plane, which is called margin. And of all the planes, the one that provides the maximum margin is considered to be the optimum hyperplane. This process gets a little bit more complex when we are dealing with a complex dataset that cannot be classified linearly.

2.7. AdaBoost

AdaBoost is a boosting classifier that boosts the accuracy of classifiers by merging several classifiers with low performance to produce a higher accuracy output. It improves the accuracy by iteratively changing the weights of the classifiers and training the dataset every iteration. (Farrugia et al., [17]). Adaboost picks a portion of the training dataset at random, then applies an iterative procedure to the dataset, initializing arbitrary weights to forecast accuracy. Improve the accuracy even further by modifying the weight in an iterative process.

2.8. LGBM classifier with Euclidean distance

Gradient boosted decision trees are used in the LGBM implementation architecture. It was designed by Microsoft's research and development team. It is a machine learning method that is commonly utilized in the prediction of studies. (Aziz et al. [27], Ahmed et al. [28], as well as Aziz et al. [4]). Because it develops vertically, LGBM is the quickest processing tree-based method when compared to other algorithms. It contains roots and leaves that may grow vertically or horizontally because it is a tree-based algorithm. As illustrated in the diagram, LGBM starts to extend across the leaf node of the tree with a significant vertical loss, that is, growing leaf-wise. Most of the algorithms, on the other hand, grow in layers or horizontally. LGBM is advantageous when results need to be computed for a huge dataset; else, it might over-fit a small dataset.

The key benefit of the proposed approach is that it is incredibly light in weight, using very little memory to compute hundreds of rows while producing extremely accurate results, as proposed by Ahamed and Education [29] (Fig. 3). The number of leaves, the learning rate, and other important factors that determine the LightGBM model's performance must be manually modified instead of being gained through training. Hyper-parameters were used to specify these parameters. Grid searching, random searching, and other traditional methods of hyper-parameter optimization are examples. Although grid searching allows for concurrent processing, it is memory intensive. Random searching aims to discover the best approximation of the function by sampling randomly in the desired range, making it easier to avoid global optima but not ensuring an optimal solution. The Euclidean distance is calculated by using the objective function's historical evaluation results, creating a probability model from them, translating the hyperparameters to the scoring probability of the objective function, and finding the optimal parameter. The parameters of Light GBM are optimized using the Euclidean distant structured estimation approach in this study. The working procedure of LGBM with Euclidean distance is shown in Fig. 4.

For the given data set $\{xi, yi\}_1^N$ The goal of training is to calculate the F* (X) function that generalizes the relationship between the input value x and the resultant value y which are used to derive outputs for untrained inputs. In other words, the loss function's expected value is minimized for the combined probability distribution of all variables (x, y).

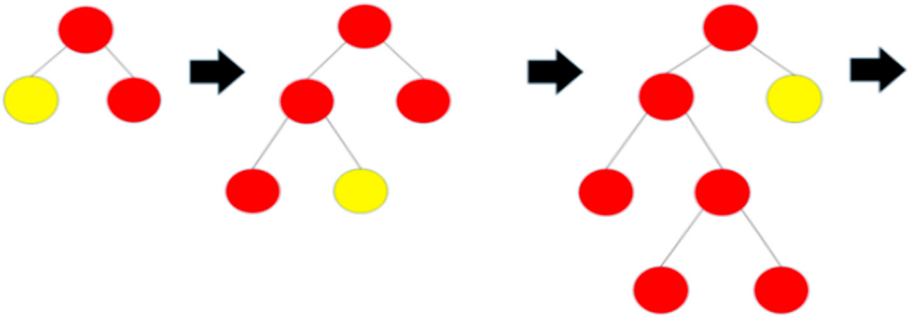

Fig. 3. Leaf wise tree growth LGBM classifier.

Gradient boosting algorithm can be mathematically implemented as below:

Step 1. Initialization model with the constant value

$$F_0(x) = \operatorname{argmin} \sum_{i=1}^{N} (y_i - \rho_0)^2 \tag{3}$$

In every iteration:

Step 2. Gradient is calculated with the help of the below equation

$$\hat{y}_i = -\left[\frac{\partial\psi(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{(m-1)}(x)} = y_i - F(x_i), i = 1\ldots N$$

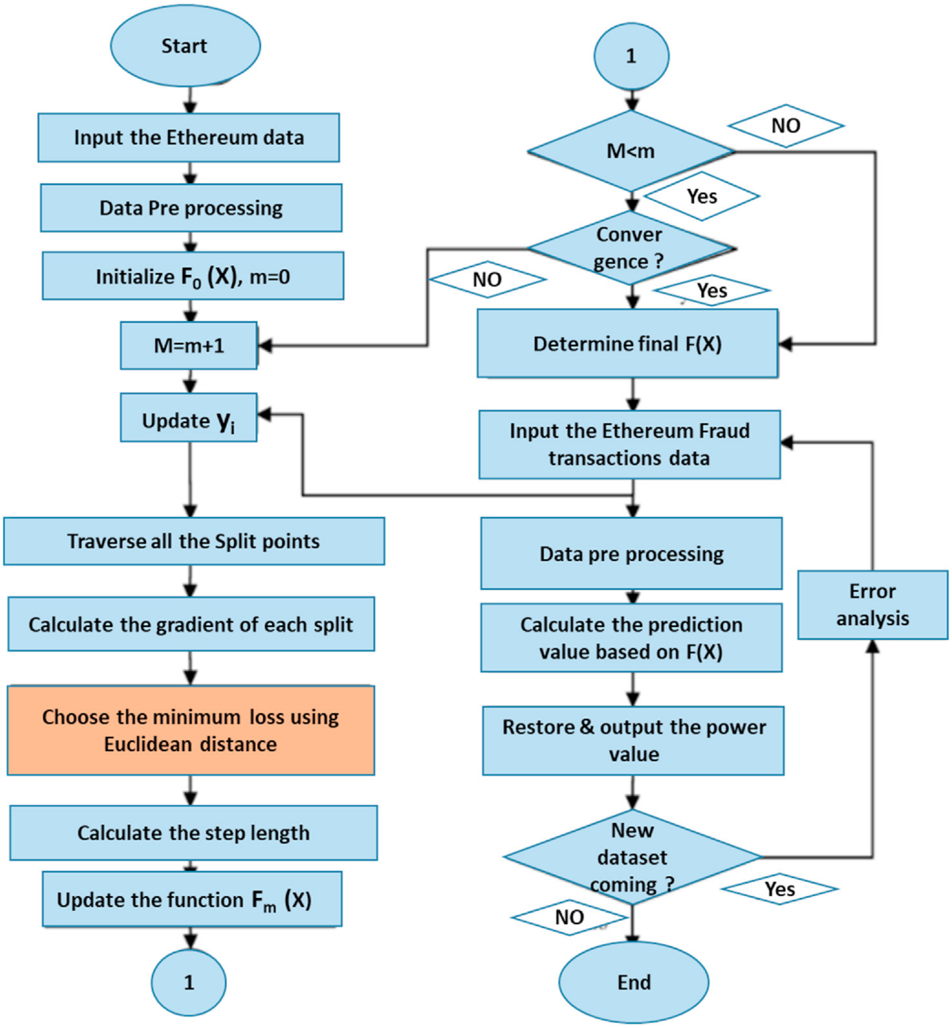
Start
Input the Ethereum data
Data Pre processing
Initialize F_0 (X), m=0
M=m+1
Update y_i
Traverse all the Split points
Calculate the gradient of each split
Choose the minimum loss using Euclidean distance
Calculate the step length
Update the function F_m (X)
1
1
M<m
NO
Yes
Conver gence ?
NO
Yes
Determine final F(X)
Input the Ethereum Fraud transactions data
Data pre processing
Calculate the prediction value based on F(X)
Restore & output the power value
New dataset coming ?
Yes
NO
End
Error analysis

Fig. 4. Iterative working procedure of LGBM with Euclidean distance.

Step 3. Under closed $h_m(x)$ to $\widehat{y_i}$ scaling, fit a base leaner closed to train the dataset $\{xi, yi\}_1^N$.

Step 4. Solve the one-dimensional below optimization equation and calculate the multiplier:

$$\rho_m = \mathrm{arg}min \sum\nolimits_{i=1}^{N} \left(y_i, F_{(m-1)}(x_i) + \rho h_m(x_i)\right) \tag{4}$$

Step 5. Update $F_m(x)$ by using below

$$\begin{aligned} F_m(x) &= F_{(m-1)}(x) \\ &\quad + \mathrm{arg}min \sum\nolimits_{i=1}^{N} \left(y_i, F_{(m-1)}(x_i) + h_m(x_i)\right) \end{aligned} \tag{5}$$

where $h_m \in H$ is a base learner function, unfortunately, in general, determining the optimum function h for an arbitrary loss function at each step is a computationally infeasible optimization issue. Therefore, we used Euclidean distant measure here for calculating the loss function for each split by equation (5).

3. Data pre-processing

The dataset comprises 9841 Ethereum transactions or rows that have been identified as fraudulent or legitimate. The Ethereum Classic (ETC) has been compiled into a Dataset, which is available on the Kaggle website. As previously indicated, the transactions table, which comprises 17 fields, is the core focus of this study. These fields are used to find abnormalities in the Ethereum network using machine learning techniques and algorithms. This dataset was first referred to by Steven et al. [17].

3.1. Experimental setup

The machine utilized for implementing the research has the following requirements as in Table 1.

The aforementioned setup is utilized to implement the full algorithm. The physical requirements are minimal, as shown in the table above, and the suggested model may be used with practically any physical machine.

Table 1. Experimental setup.

OS	Windows 10
RAM	8 GB
GPU	4 GB
IDE	Python

3.2. Pre-processing

The dataset is unbalanced, which may affect the model's accuracy. To balance the classes, the minority upscale should be resampled to match the frequency with the majority of the dominant class. The data source originally consisted of 9841 items, with 7662 having '0' as the outcome, as seen in Fig. 5. Tainted records like transactions with zero values cannot be comprehended as suspicious or legit transactions, so we eliminate them, finally, there are a total of 2179 items in the dataset (Figs. 6 and 7).

The heatmap in Fig. 7 shows that there are multiple Ethereum transactions with missing attribute values after computing the percentage of missing values of columns in rows and visualizing the heatmap. Hence, removing the features with maximum null values. Fig. 7, describes which attributes are missing in the transactions, which makes this data unable to be used for further processing.

3.3. Distribution of features

Figure 8 illustrates the distribution of characteristics.

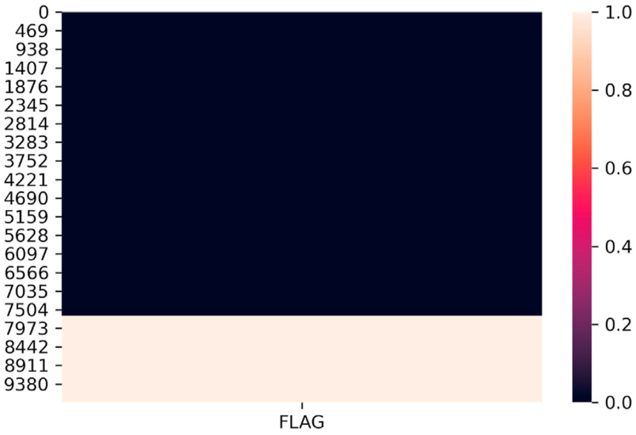

Fig. 5. Used Ethereum dataset distribution of values having '1' & '0'.

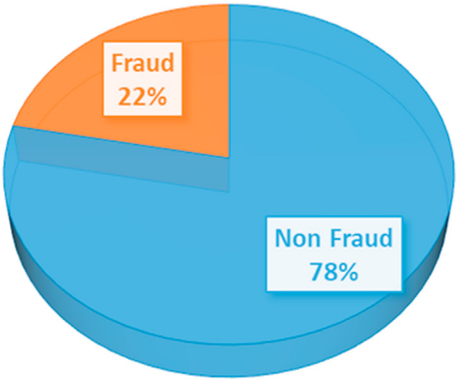

Fig. 6. Percentage of fraudulent value in the used dataset.

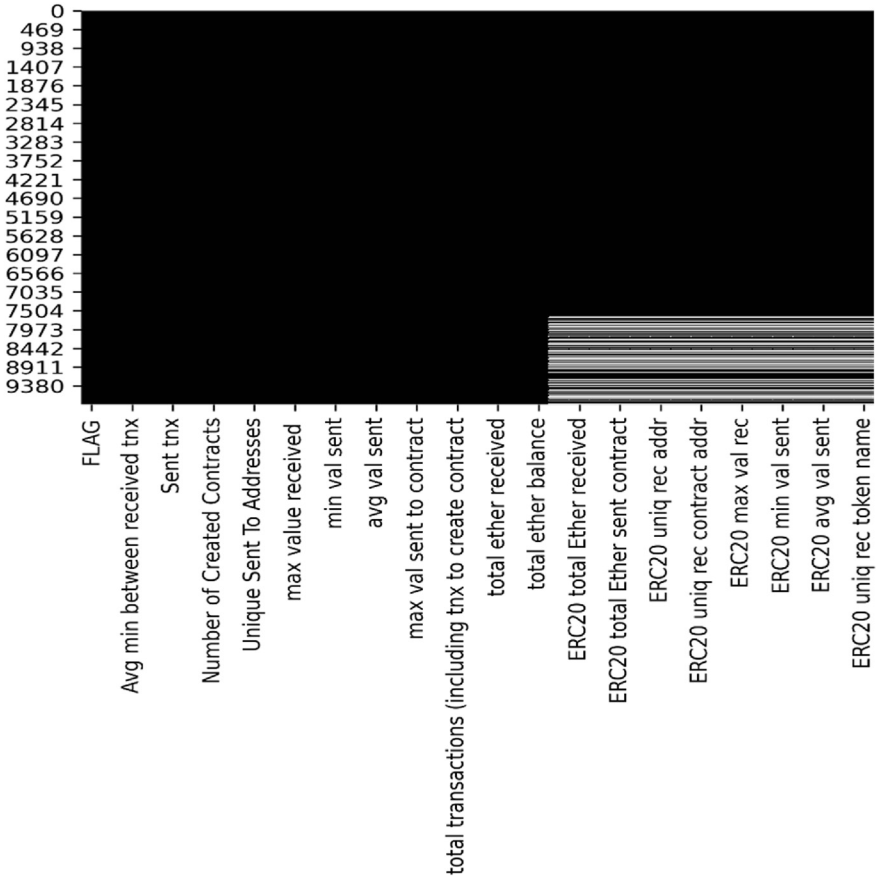

Fig. 7. Heatmap of missing values in the used dataset.

'ERC20 Uniq sent addr.1' and min value sent to contract': These two variables have a lot of null values, as can be seen. As a result, both features will be eliminated because they are no longer relevant to the model.

3.3.1. Splitting the dataset

The dataset is split into two parts: a training dataset that the model uses to adapt to the data and a testing dataset that confirms and supports the trained model's validity. The dataset split in this research study is 4:1, or 80 percent −20 percent for training and testing, respectively (Fig. 8).

3.4. Normalizing the training features

The dissemination of features following log transformation is evaluated with the help of the function, power transform.

3.5. Management for the Imbalance data

SMOTE (Synthetic Minority Oversampling Approach) is the ML technique for classification balancing that involves oversampling.

Unbalanced categorization is the frequent common issue encountered while training a model. Instead of deleting plentiful entries from the dataset, this strategy duplicates values in the minority class. These repeated values, however, do not provide any new information.

BEFORE OVERSAMPLING Frauds: 1757, Non-frauds: 6115.

AFTER OVERSAMPLING: Frauds: 6116, Non-frauds: 6115.

4. Experimental results and discussions

With the help of the sklearn library of python, the dataset has been bifurcated into two sets:

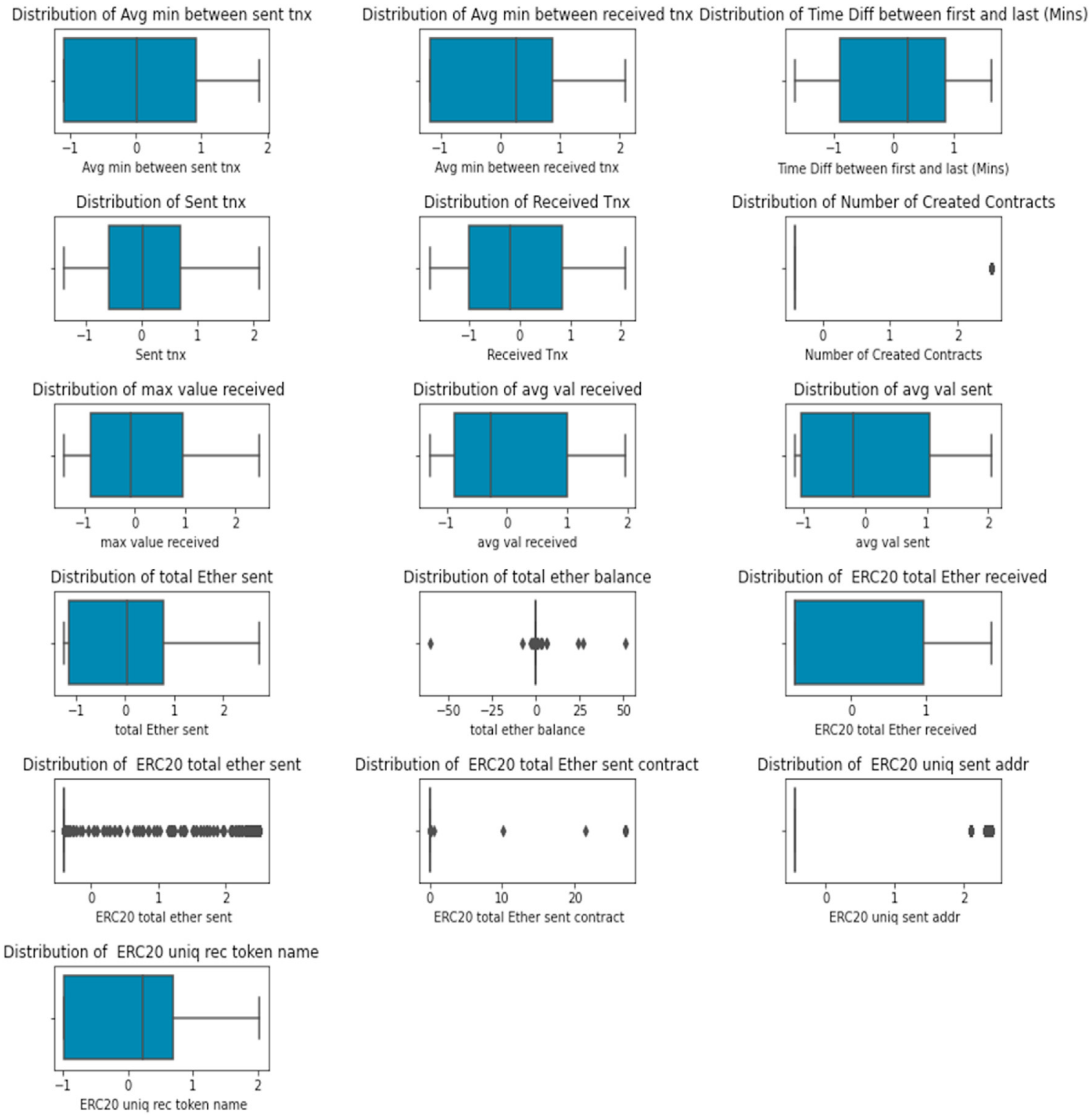

Fig. 8. Boxplots distribution of features after log transformation.

testing and training sets, each with a 20%−80% ratio. The selection of hyperparameters is crucial in modelling. In Light GBM, there are several hyperparameters to choose from. Only the elements that have a substantial impact on model performance were selected for parameter tuning to increase real-time fault detection performance. Table 2 lists all the important parameters which have been employed for the model. The suggested framework is evaluated in terms of metric accuracy, and the F1-score and precision are used to compare scores.

Table-2. Hyper-parameter setting of MODIFIED LGBM.

LGBM Classifier	
num_leaves	2000
min_data_in_leaf	120
max_depth	13
feature_fraction	0.5
bagging fraction	0.8

Table 3 shows the results in the form of training, testing accuracy as well as F_1 score. The confusion matrix of a different model with different classifiers provided recall and precision.

In addition, Fig. 9 suggests that the modified LGBM performs much better without compromising other parameters and overfitting by providing the best possible accuracy, along with Random Forest. In contrast, among the models, logistic regression and SVC had the lowest accuracy. The confusion matrix for the modified LGBM

Table 3. Comparison of performance of the proposed algorithm with popular published algorithms.

	Training Accuracy	Testing Accuracy	Precision	Recall	F1 Score
MLP Classifier	96.59	95.3	92.75	87.31	89.08
Logistic Regression	86.9	84.92	92.64	88.53	56.85
Random Forest	98.93	98.26	98.54	90.78	94.43
KNN	95.7	94.84	90.31	83.1	87.71
XGBoost	98.4	97.99	97.07	93.5	95.38
SVC	87.18	85.12	99.77	92.28	56.47
ADAboost	98.09	97.94	95.54	93.25	95.29
Modified LGBM Classifier	99.29	99.17	97.47	93.75	95.62

classifier after hyper-parameter tuning is demonstrated in Fig. 10 below.

4.1. Best model evaluation

As shown in Table 3, we got the highest training and testing accuracy after we enhanced our LGBM Classifier. That is why, with the aid of hyper-parameter tuning, we will further enhance our LGBM parameters. It controls numerous features of the model, including data overfitting and underfitting. As a result, we used randomized hyper-parameter tuning for our LGBM classifier and obtained high accuracy with optimized parameters. Table 2 shows the optimum parameter values for the best outcome.

Figure 9 depicted that after applying hyper-parameter tuning, we were able to optimize the code as a result, our accuracy increased to 99.17%.

4.2. Importance features

To comprehend the importance of each feature of the optimal model. The graph shows that the most important features in identifying fraudulent transactions are "Difference in time among Initial and final (Minutes)" and " Addresses that are unique have been recognized " (Fig. 11).

4.3. ROC of different models using different classifiers

In addition to the F1-score, training attesting accuracy, Recall, and Precision is used in the above section. We continued to use Receiver Operating Characteristic (ROC) has been used to calculate the performance by each model, and respective results are shown in Fig. 12. The ROC of modified LGBM is 98.27 in terms of the area under the curve, which means the classification accuracy of modified LGBM is 98.27%. It can be proved that modified LGBM performs well in the classification task.

We can see in Fig. 12, modified LGBM has a relatively higher performance than other algorithms. The ROC of different models also signifies that the modified LGBM classifier produced better results than other popular algorithms. Therefore, the modified LGBM with hyper-parameter setting is

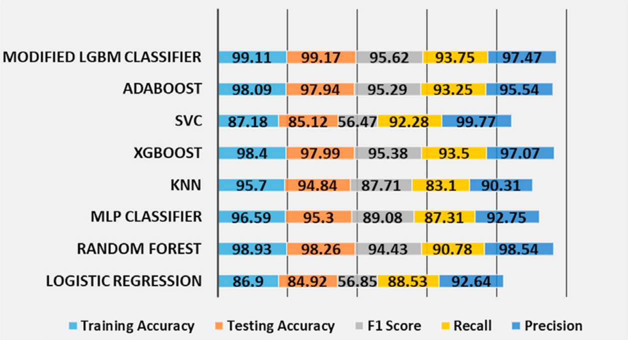

Fig. 9. Comparison of evaluation metrics of different methods.

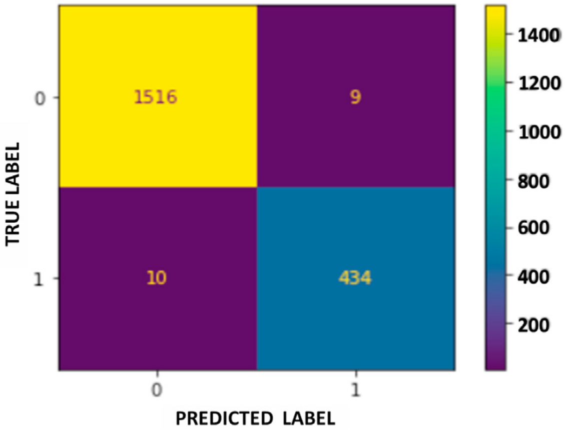

Fig. 10. Confusion matrix for modified LGBM classifier after hyper-parameter tuning.

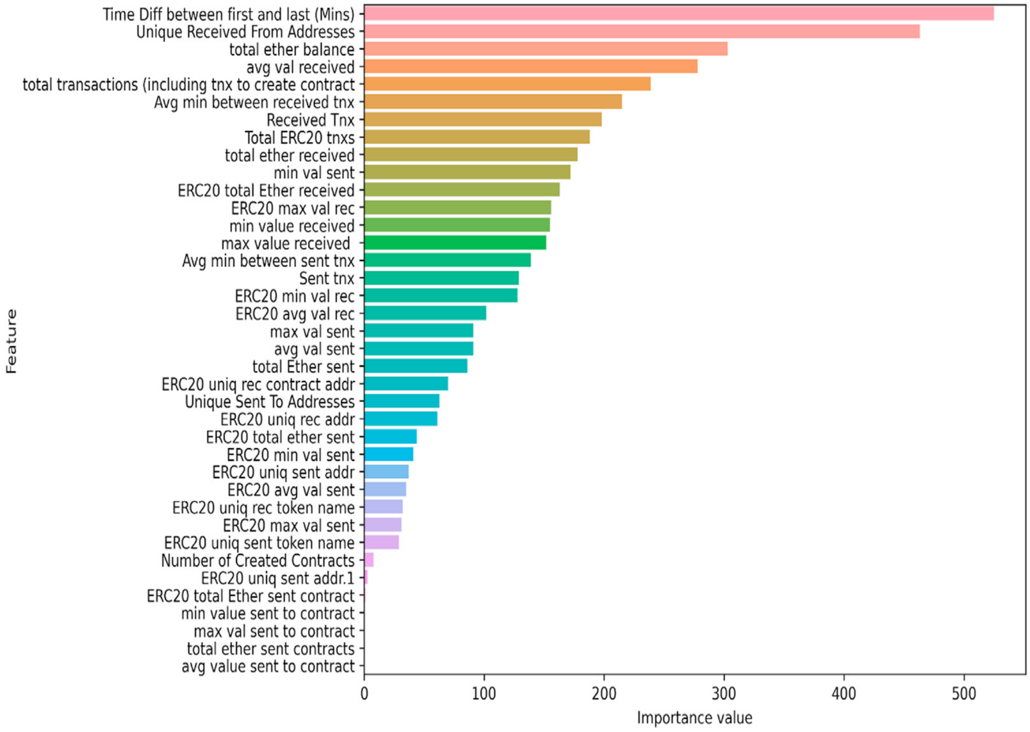

Fig. 11. Distribution of features by importance.

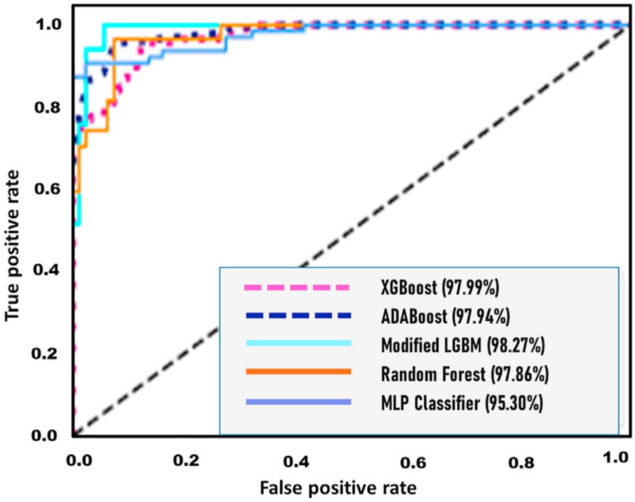

Fig. 12. ROC of Random Forest, MLP Classifier, XGBoost, ADAboost, and modified LGBM based model.

a powerful boosting framework for the detection of Ethereum fraud transactions.

5. Conclusions

For identifying Ethereum fraudulent transactions, this paper proposed a modified LGBM classifier. For the Ethereum fraudulent transactions, a significant experiment was conducted to assess the suggested technique's performance using several machine learning algorithms. The accuracy of eight different machine learning algorithms, including Random Forest, LGBM Classifier, MLP Classifier, and Logistic Regression, was the major emphasis of the study. There are 9841 variables and 43 attributes in the dataset that was used to train and test the models. After preprocessing and selecting characteristics, the bulk of the algorithms worked well. However, the data demonstrate that improved gradient boosting approaches and RF surpassed some of the most complex machine learning systems, with maximum accuracy of 99.17 percent and 98.26 percent, respectively.

It was also determined that using certain parameters derived from the LGBM model's hyperparameter tuning, accuracy improved to the greatest achievable result of 99.17 percent. To detect fraudulent transactions, the suggested model incorporates the LGBM technique. This model produced reliable findings, but it has one flaw: for LGBM to function effectively, the dataset must be significantly large. Elephant Herding Optimization (EHO), Earthworm Optimization Algorithm (EWA) [34], Monarch Butterfly Optimization (MBO) [35], Moth Search Algorithm (MSA), and Slime Mould Algorithm (SMA) [36] are some of the most representative computational intelligence algorithms that can be used to evaluate the problems. Trends in Ethereum transactions can be noticed, which should be studied for future research. It's also possible to improve on this research and come up with a machine learning model that's much more accurate across all dataset sizes.

Conflict of interest

The authors declare no conflict of interest.

Acknowledgements

The authors would like to thank the editors and the anonymous reviewers for their valuable comments and suggestions, which have helped immensely in improving the quality of the paper.

References

[1] L. Clohessy, H. Treiblmaier, T. Acton, N. Rogers, Antecedents of blockchain adoption: an integrative framework, Strat Change 29 (2020) 501–515, https://doi.org/10.1002/jsc.2360.

[2] X. Li, A.B. Whinston, Analyzing cryptocurrencies, Inf Syst Front 22 (2020) 17–22, https://doi.org/10.1007/s10796-019-09966-2.

[3] J. Liu, A. Serletis, Volatility in the cryptocurrency market, open econ, Rev 30 (2019) 779–811, https://doi.org/10.1007/s11079-019-09547-5.

[4] R. Aziz, M.F. Baluch, S. Patel, A.H. Ganie, LGBM: a machine learning approach for Ethereum fraud detection, Int J Inf Technol 14 (2022) 1–11, https://doi.org/10.1007/s41870-022-00864-6.

[5] F. Leal, A.E. Chis, G.V. Horacio, Multi-service model for blockchain networks, Info. Proc. Mngt. 58 (2021) 102519–102525, https://doi.org/10.1016/j.ipm.2021.102525.

[6] A. Panarello, N. Tapas, G. Merlino, F. Longo, A. Puliafito, Blockchain and IoT integration: a systematic survey, Sensors 18 (2018) 2569–2575, https://doi.org/10.3390/s18082575.

[7] Z. Zhao, Yuan Liu, A block chain based identity management system considering reputation, in: 2019 2nd Int. Conf. Inf. Syst. Comput. Aided Educ, IEEE, 2019, pp. 32–36, https://doi.org/10.1109/ICISCAE48440.2019.221582.

[8] A. Kabašinskas, K. Šutienė, Key roles of crypto-exchanges in generating arbitrage opportunities, Entropy 23 (2021) 447–455, https://doi.org/10.3390/e23040455.

[9] A. Brauneis, R. Mestel, E. Theissen, What drives the liquidity of cryptocurrencies? A long-term analysis, Finance Res Lett 39 (2021) 1010531–1101537, https://doi.org/10.1016/j.frl.2020.101537.

[10] E. Jung, M.L. Tilly, A. Gehani, Y. Ge, Data mining-based ethereum fraud detection, in: Intl. Conf. On blockchain., IEEE, 2019, pp. 266–273, https://doi.org/10.1109/Blockchain.2019.00042.

[11] M. Bartoletti, S. Carta, T. Cimoli, R. Saia, Dissecting ponzi schemes on ethereum: identification, analysis, and impact, Future Generat Comput Syst 102 (2020) 259–277, https://doi.org/10.1016/j.future.2019.08.014.

[12] W. Chen, Z. Zheng, E.C.H. Ngai, P. Zheng, Y. Zhou, Exploiting blockchain data to detect smart ponzi schemes on ethereum, IEEE Access 7 (2019) 37575–37586, https://doi.org/10.1109/ACCESS.2019.2905769.

[13] M. Bartoletti, B. Pes, S. Serusi, Data mining for detecting bitcoin ponzi schemes, in: Crypto valley conf. On blockchain technology (CVCBT), IEEE, 2018, pp. 75–84, https://doi.org/10.1109/CVCBT.2018.00014.

[14] M. Vasek T. Moore, Analyzing the bitcoin ponzi scheme ecosystem, in: Intl. Conf. On financial cryptography and data security, Springer, 2018, pp. 101–112, https://doi.org/10.1007/978-3-662-58820-88.

[15] K. Ajay, K. Abhishek, P. Nerurkar, M.R. Ghalib, A. Shankar, X. Cheng, Secure smart contracts for cloud based manufacturing using Ethereum blockchain, Trans Emerg Telecommun 13 (2020) 4121–4129, https://doi.org/10.1002/ett.4129.

[16] Teng Hu, Xiaolei Liu, Ting Chen, Xiaosong Zhang, Xiaoming Huang, Weina Niu, Jiazhong Lu, Kun Zhou, Yuan Liu, Transaction-based classification and detection approach for ethereum smart contract, Inf Process Manag 58 (2021) 102457–102462, https://doi.org/10.1016/j.ipm.2020.102462.

[17] R. Tan, Q. Tan, P. Zhang, Z. Li, Graph neural network for ethereum fraud detection, in: IEEE intl. Conf. On big knowledge, ICBK, 2021, pp. 78–85. https://doi: 10.1109/ICKG52313.2021.00020.

[18] Q. Yuan, B. Huang, J. Zhang, J. Wu, H. Zhang, X. Zhang, Detecting phishing scams on ethereum based on transaction records, in: IEEE intl. Symposium on circuits and systems (ISCAS), IEEE, 2020, pp. 1–5, https://doi.org/10.1109/ISCAS45731.2020.9180815.

[19] R.F. Ibrahim, A. Mohammad Elian, M. Ababneh, Illicit account detection in the ethereum blockchain using machine learning, in: IEEE intl. Conf. On information technology, ICIT, 2021, pp. 488–493. https://doi: 10.1109/ICIT52682.2021.9491653.

[20] W.J. Tsaur, J.C. Chang, C.L. Chen, A highly secure IoT firmware update mechanism using blockchain, Sensors 22 (2022) 530–549, https://doi.org/10.3390/s22020530W.

[21] Z. Chen, E.C.H. Zheng, P. Ngai, Y. Zheng, Exploiting blockchain data to detect smart ponzi schemes on ethereum, IEEE Access 7 (2019) 37575–37586, https://doi.org/10.1109/ACCESS.2019.2905769.

[22] R. Aziz, H. Aftab, S. Prajwal, P. Kumar, Machine learning-based soft computing regression analysis approach for crime data prediction, Karbala Int J Mod Sci 8 (2022) 1–19, https://doi.org/10.33640/2405-609X.3197.

[23] L. Chen, J. Peng, Y. Liu, J. Li, F. Xie, Z. Zheng, Phishing scams detection in ethereum transaction network, ACM Trans Internet Technol 21 (2020) 1–16, https://doi.org/10.1145/3398071.

[24] H. Arimura, M. Soufi, H. Kamezawa, K. Ninomiya, M. Yamada, Radiomics with artificial intelligence for precision medicine in radiation therapy, J Radiat Res 60 (2019) 150–157, https://doi.org/10.1093/jrr/rry077.

[25] S. Sreejith, S. Rahul, R. Jisha, A real time patient monitoring system for heart disease prediction using random forest algorithm, in: Advances in signal processing and intelligent recognition systems, Springer, 2016, pp. 485–500, https://doi.org/10.1007/978-3-319-28658-7_41.

[26] R. Aziz, C.K. Verma, N. Srivastava, A fuzzy based feature selection from independent component subspace for machine learning classification of microarray data, Genomics data 8 (2016) 4–15, https://doi.org/10.1016/j.gdata.2016.02.012.

[27] R. Aziz, C.K. Verma, N. Srivastava, Novel machine learning approach for classification of high-dimensional microarray data, Soft Comput 23 (2019) 13409–13421, https://doi.org/10.1007/s00500-019-03879-7.

[28] B.S. Ahamed, M.S. Arya, LGBM classifier based technique for predicting type-2 diabetes, Euro J Mol Clin Med 8 (2021) 454–467.

[29] B.S. Ahamed, M. Education, Prediction of type-2 diabetes using the LGBM classifier methods and techniques, Turkish J Comp Math Edu 12 (2021) 223–231.

[30] R. Aziz, C.K. Verma, M. Jha, N. Srivastava, Artificial neural network classification of microarray data using new hybrid gene selection method, Intl J Data Mining Bioinformatics 17 (2017) 42–65.

[31] R. Aziz, C.K. Verma, N. Srivastava, A weighted-SNR feature selection from independent component subspace for nb classification of microarray data, Int J Adv Biotechnol Res 6 (2015) 245—255.

[32] R. Aziz, N. Srivastava, C.K. Verma, T-independent component analysis for SVM classification of DNA-microarray data, Int J Bioinf Res 6 (2015) 305—312.

[33] N.P. Desai, M.F. Baluch, A. Makrariya, R.M. Aziz, Image processing model with deep learning approach for fish species classification, Turk J Comput Math Educ 13 (2020) 85—99.

[34] G.G. Wang, S. Deb, L.D.S. Coelho, Earthworm optimization algorithm: a bio-inspired metaheuristic algorithm for global optimization problems, Int J Bio-Inspired Comput 12 (2018) 1—22.

[35] Y. Feng, S. Deb, G.-G. Wang, A.H. Alavi, Monarch butterfly optimization: a comprehensive review, Expert Syst Appl 168 (2021) 114418, https://doi.org/10.1016/j.eswa.2020.114418.

[36] S. Li, H. Chen, M. Wang, A.A. Heidari, S. Mirjalili, Slime mould algorithm: a new method for stochastic optimization, Future Generat Comput Syst 111 (2020) 300—323, https://doi.org/10.1016/j.future.2020.03.055.