

Detection of Digital Currency Fraud through a Distributed Database Approach and Machine Learning Model

Faisal Ghazi Abdiwi ¹

¹*Department of Computer Science & Information Systems, Al Mansour University College Baghdad, Iraq*

Abstract – The world is witnessing a noticeable increase in financial exchange in digital currencies such as Bitcoin, Ethereum, and others, as transactions in electronic markets have begun to rise recently, which increases the difficulty of maintaining security and trust in decentralized financial systems that use distributed databases and the technologies that interact with them in Ethereum networks, blockchain, etc. This study presents a hybrid model based on the PyCaret library and includes 12 machine learning classifiers, with the aim of identifying fraudulent activities in Bitcoin transactions and enhancing the security of Ethereum networks and blockchain technology. The results reveal the effectiveness of different models in identifying fraudulent activities on the Ethereum network through a comprehensive performance comparison. The classifiers that showed the highest accuracy scores, which ranged from 0.9814 to 0.9862, were the Random Forest classifier, the visual gradient boosting machine, and the additive tree classifier. It is important to note that both Gradient Boosting Classifier and K Neighbors Classifier performed well, with accuracies above 0.96 and AUC scores above 0.99.

However, some models, such as Naive Bayes, showed lower accuracy and AUC scores, suggesting that they have limitations in terms of accurately detecting fraudulent transactions.

These results highlight the importance of choosing appropriate machine learning models for fraud detection tasks in general, with ensemble techniques such as Extra Trees and Random Forest showing great promise in this regard.

Keywords – Distributed databases, machine learning, Pycarte, blockchain, Ethereum.

1. Introduction

Cryptocurrencies like Ethereum and Bitcoin have seen a huge rise in demand, especially among large institutions, since their inception. One of its primary goals is to decentralize the management of financial markets, allowing users to control their transactions and data. Ethereum, developed by Vitalik Buterin, acts as a cryptocurrency transfer system for users to send cryptocurrencies to anyone for a small fee. Ethereum also provides a simplified approach to digital transactions, and can be accessed by individuals all over the world. Ethereum, like other cryptocurrencies, is powered by the blockchain network, a distributed, decentralized public database that keeps track of and validates each transaction. Since cryptocurrencies are decentralized, no party can control the Ethereum network, maintaining confidentiality and protecting companies. However, as the proliferation of Ponzi schemes on Ethereum shows, it can also provide opportunities for dishonest activity. Ponzi schemes that represent real investment opportunities have become very popular. These schemes operate by using money from later investors to compensate early investors. This creates a pyramid-shaped structure where profits flow to the top and losses pile up to investors at the bottom. Ponzi schemes almost always fail when they cannot sustainably recruit new investors, even with their initial returns. Many investigations have been conducted using various AI classifiers and techniques for fraud detection in Ethereum.

The decentralized and anonymous structure of digital currencies like Bitcoin and Ethereum makes them vulnerable to fraud, which is a big problem for digital trading markets.

DOI: 10.18421/TEM134-37

<https://doi.org/10.18421/TEM134-37>

Corresponding author: Faisal Ghazi Abdiwi,
*Department of Computer Science & information systems,
Al Mansour University College, Baghdad, Iraq.*


Email: faisal.ghazi@muc.edu.iq

Received: 05 May 2024.

Revised: 22 September 2024.

Accepted: 05 November 2024.

Published: 27 November 2024

 © 2024 Faisal Ghazi Abdiwi; published by UIKTEN. This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivs 4.0 License.

The article is published with Open Access at <https://www.temjournal.com/>

Fraud through false offers and projects promising unreasonably high returns is one of the biggest concerns. Scammers exploit people's constant desire for quick and easy profits to lure victims out of their money.

In addition, another risk is cyber hacks and thefts targeting cryptocurrency exchanges and currency wallets. Hackers try to infiltrate these platforms to steal users' digital funds, leading to huge losses for investors and traders. Therefore, the research proposes a system for maintaining the security of digital exchange markets using machine learning techniques. The study aims to evaluate the effectiveness of the proposed model in detecting fraudulent activities on the Ethereum blockchain by identifying anomalies in transaction data. A thorough comparison using four machine learning techniques was carried out, such as decision tree and Random Forest algorithms, among others, to assess their performance based on several metrics, with accurate serving as the main factor [1], [2], [3], [4].

2. A Distributed Database (DDB)

A distributed database (DDB) includes several interconnected databases spread across a computer network. Managing this distributed environment is the role of a distributed database management system (DDBMS), which ensures users' seamless access to distributed data. The integration of a distributed database and management system into a group of machines that do not have shared memory acts as a cohesive unit and appears as a single entity to the user. The motivations for using distributed databases stem from various factors. First, some applications naturally fit into a distributed architecture because of the distinction between global and local scales. For example, a bank that has local branches and a central office would naturally distribute its database across these local locations and the central office. A distributed database system's architecture is depicted in Figure 1 [5].

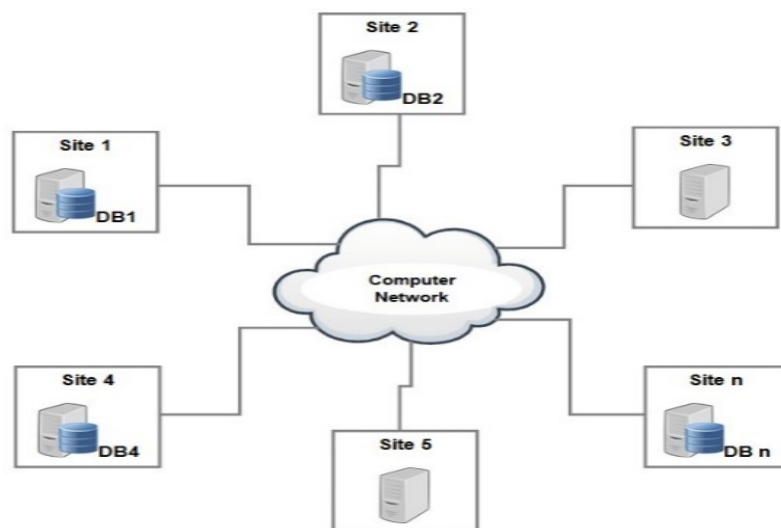


Figure 1. Architecture of a distributed database system

The distributed architectural concept is embodied in blockchain technology, as shown in Figure 2, where blockchain networks create distributed databases by distributing data among several nodes in a decentralized manner. Each node keeps a copy of the complete blockchain ledger in order to offer redundancy and fault tolerance.

Due to its dispersed nature, which eliminates the necessity of a central authority, this increases network participants' transparency and trust. The decentralized aspect of Blockchain, which is similar to a distributed database system, is further highlighted by the network's consensus mechanism procedure, which ensures data integrity and immutability [6].

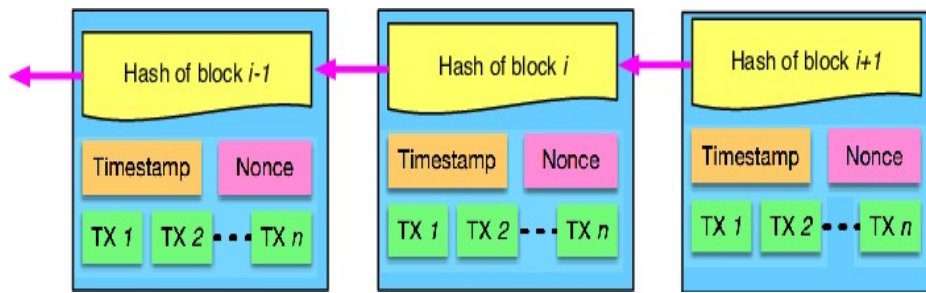


Figure 2. Blockchain blocks

Ethereum is a software platform that disseminates copies of smart contracts to a wide global network of users via blockchain technology. Ethereum being a digital currency, makes the exchange of value possible anywhere in the world without relying on a single entity (due to its decentralized nature). But as e-commerce expands, so do the fraudulent practices of phishing and money laundering, which pose serious risks to business security. Ethereum was created in 2015 with the goal of building a global decentralized computing platform that uses the security of blockchain technology and opens it up to a variety of uses that go beyond digital transactions. The Ethereum network has potential beyond imagination. It allows developers to create and implement applications ranging from sophisticated databases to financial instruments. One of the core components of the Ethereum network is smart contracts, which run on a decentralized blockchain similar to Bitcoin.

These contracts are used to automate agreements between parties without the need for intermediaries, allowing applications to be executed securely and without central control. These contracts are powered by Ethereum’s native cryptocurrency, known as ether (ETH), and the costs paid for using it are known as “gas.” Like Bitcoin, the Ethereum network is open-source and decentralized, allowing anyone with an internet connection to join. With the release of Ethereum 2.0 in September 2022, a major upgrade to the platform was made with the aim of improving efficiency and scalability. The Proof of Stake (PoS) consensus process has replaced the energy-intensive Proof of Work (PoW) mechanism. In addition to addressing issues related to high fees and network congestion, which in turn contributes to improving the long-term sustainability and benefit of Ethereum. By 2024, Ethereum has become the dominant platform for blockchain applications, with a market value approaching \$265 billion, although it faces competition from emerging smart contracts.

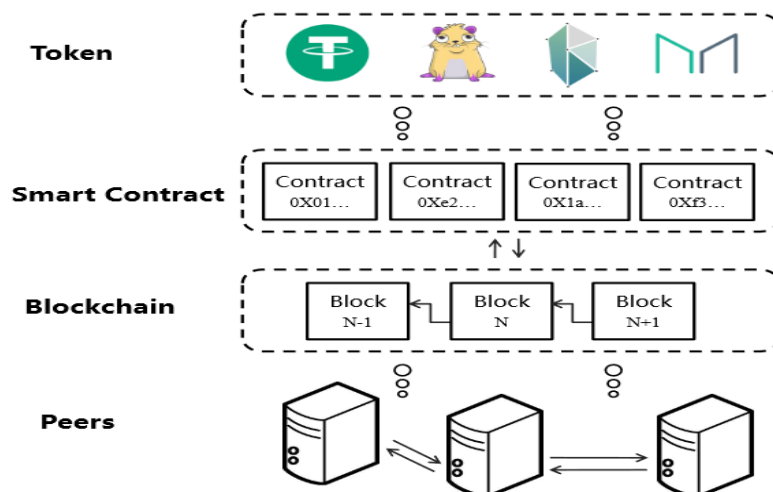


Figure 3. Overview of Ethereum blockchain

Figure 3 illustrates the structure of the Ethereum. The Ethereum blockchain's architecture is depicted in Figure 3, with peers, the blockchain, smart contracts, and tokens arranged in a top-to-bottom fashion [7].

A blockchain is basically a chain-like data structure made up of blocks connected one after another and controlled by each peer in a peer-to-peer blockchain network.

Each block produced by a peer, also known as a miner, includes the hash value of the previous block along with the confirmed transactions. The block is independently verified by other peers after it is created. Transactions in a block are considered completed when their authenticity is verified and confirmed by the majority of participants in the blockchain network. Since all peers verify the validity of transactions, redundant calculations and storage increase the trustworthiness of the data, and as a result, each peer trusts the entire blockchain. Researchers' access to the complete blockchain data set is facilitated by the comprehensiveness of blockchain data within each permissionless blockchain peer. The blockchain data set is valuable, covering mining operations and users. Transaction records, for example, capture transactions between different business organizations and provide insight into real-world economic systems such as money transfers. Furthermore, blockchain data analysis is becoming more important in predicting economic trends due to the growth of blockchain users and transactions, especially in Bitcoin and Ethereum. An idea that predates blockchain technology called smart contracts shows promise as a tool to change industries. Smart contracts are published or activated through blockchain transactions, ensuring the reliability of the contracts. Ethereum offers Turing-complete smart contracts (while current blockchain platforms allow only basic smart contracts). These contracts are executed by the Ethereum Virtual Machine (EVM), which uses smart contract actions to read and write states saved in the key-value database. Notably, token issuance on top of Ethereum is made possible by two standard protocols (or templates) for smart contracts. This is done by defining common variables, functions, and interfaces for smart contracts. These protocols allow users to issue tokens, or cryptocurrencies, without the need for a central authority. Ethereum hosts several tokens such as USDT, Cryptokitties, Kyber, and MarkerDAO, expanding the financial ecosystem of Ethereum. Ethereum accommodates several clients including Go-Ethereum (Geth) and Parity. Geth and Parity provide JSON-RPC interfaces to interact with the Ethereum blockchain, with Parity being better equipped to retrieve data due to the improved interface design, ensuring accurate and efficient data retrieval [8], [9], [10], [11], [12], [13].

3. Related Work

Works related to the subject of scientific research are of great importance, as they provide a deep idea of what researchers have achieved in the problem and what its strengths and weaknesses are. This, in turn, contributes to refining the proposed idea.

Therefore, this section highlights the most important findings of previous work in the field of user behavior analysis and financial fraud detection in digital currency networks:

In the approach presented by Rabei Mushir Aziz and colleagues [15], a hybrid machine learning approach was trained which consisted of a set of classifiers such as ADABOOST, K-Nearest Neighbors (KNN), support vector machine (SVC), XGBoost, Random Forest (RF) and multi-layer perceptron (MLP), and regression [15]. The results of this study showed the superiority of the updated LGBM model with 99.17% accuracy and RF with 98.26% accuracy to address financial anonymity fraud cases such as ICO exits, phishing, and fraudulent scams in the Ethereum network. Presented by Amer Salam *et al.*, the study seeks to provide an intelligent model consisting of, such as K-Nearest Neighbor, Random Forest, and XGBoost to distinguish between the characteristics of a legitimate account and a fraudulent account [15]. A total of 4,681 cases—2,179 fraudulent accounts and 2,502 legitimate accounts—were taken from the dataset. The average accuracy of XGBoost, RF, and KNN approaches was 96.68% and 94.88%, with corresponding AUCs of 0.995 and 0.99 [16]. In response to the monitoring and detection of fraudulent activities and cryptocurrency crimes in 2021, and the expected increase in blockchain transaction through, with a particular focus on the Ethereum blockchain, researchers presented a model to detect blacklisted addresses within the Ethereum blockchain. Next, a graph of Ethereum transactions was proposed, which contributed to extracting the most important features such as PageRank. The model recorded over 97% accuracy in predicting blacklisted addresses. Hamed Hassan Branto *et al.* [17] proposed a method based on blockchain technology and smart contracts to develop a machine learning algorithm that aids in detecting commercial fraud while ensuring data privacy through the blockchain network. The proposed model was tested across eight additional updates. For blockchain difficulty levels below five, the experiment showed a positive relationship between time mining and difficulty level. The testing accuracy of the proposed model was 98.93% and the Fbeta score was 98.22%. Also, Norsingha Tripathi *et al.* [18] addressed the issue of increasing crimes in the Ethereum network. The study used a model proposed by a group of machine learning algorithms such as (AdaBoost), logistic regression (LR), Random Forest (RF), k-nearest neighbors (KNN), and extreme gradient boosting (XGBoost). Experimental tests were conducted on the Ethereum financial fraud dataset. The classification models showed different behavior in the test trials. The recorded accuracy of the LRmodel was (0.933).

RF recorded high accuracy (0.929) while KNN classifier showed good accuracy (0.807), and AdaBoost classifier accurately (0.90). The XGB classifier had the highest accuracy (0.946). In protecting healthcare systems from cyber fraud, researchers in the study [19] presented a decision tree-based machine learning model to analyze healthcare data and organize a smart contract on the Ethereum and blockchain networks to detect user behavior and prevent financial fraud. The proposed model recorded an excellent classification accuracy of 97.96%. The study confirmed that smart contracts on the Ethereum network and blockchain have become much better at detecting fraud. Finally, with the rise in cybercrime in cryptocurrency markets, Tur Tahir *et al.* [19] developed an advanced model for fraud detection in blockchain transactions, also focusing on the Ethereum network.

The approach proposes using machine learning ensemble methods with hard voting (ML). The proposed ensemble model achieved 99% accuracy in experimental tests. In addition, the study focuses on using Explainable Artificial Intelligence (XAI) to improve the transparency of AI-based fraud detection systems.

4. Proposed Methodology

In this study, an artificial intelligence-based model for analyzing and detecting financial fraud in digital transactions within the Ethereum network was reviewed. Figure 4 shows the proposed method, which includes a set of stages represented by analyzing a data set related to encrypted electronic currencies and identifying important features and characteristics used in experimental testing.

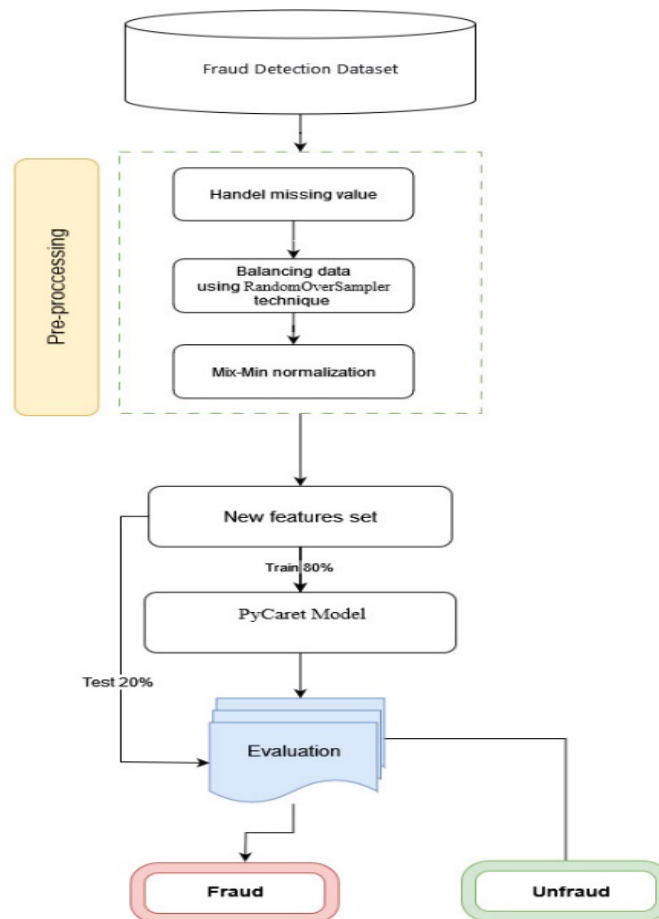


Figure 4. The proposed systems

Open source AI technologies are demonstrated for analyzing data and detecting financial fraud. The paper will elaborate on this approach and emphasize the critical role of the proposed model in cryptocurrency networks vulnerable to financial fraud.

4.1. Dataset

To carry out experiments in order to evaluate the proposed model, a data set on financial fraud of digital currencies in the Ethereum network was used. The dataset consists of 9,481 records divided into 50 features, details of which are shown in Table 1 [18].

Table 1. The features in the dataset

Feature name	Details
Index	A unique identifier for each transaction entry.
Address	The Ethereum account address involved in the transaction.
FLAG	Indicates whether the transaction is classified as fraudulent or valid.
Minimum Average Between Transmissions Sent	The average number of minutes that elapses between transactions submitted to an account.
Avg_min_between_received_tnx	The average number of minutes that pass between an account's transactions.
Time_Diff_between_first_and_last(Mins)	The time difference between the first and last transaction related to the account, and is measured in minutes.
Sent_tnx	The total number of regular transactions sent.
Received_tnx	The total number of regular transactions received.
Number_of_contracts_created	The total number of contracts created by the account.
Unique_Received_From_Addresses	The entire list of unique addresses that the account received transactions from.
Unique_Sent_To_Addresses20	Total unique addresses to which the account sent transactions.
Min_Value_Received	The minimum value of ether received by the account.
Max_Value_Received	The maximum value of Ether that the account will ever receive.
Avg_Value_Received5Average	The value of ether the account has ever received.
Min_Val_Sent	The minimum value of ether sent via the account.
Max_Val_Sent	The maximum value of ether sent via the account.
Avg_Val_Sent	Average value of ether sent via the account.
Min_Value_Sent_To_Contract	The minimum value of ether sent to the contract by the account.
Max_Value_Sent_To_Contract	The maximum value of ether sent to the contract by the account.
Avg_Value_Sent_To_Contract	Average value of ether sent to contracts by account.
Total_Transactions(including_Tnx_to_Create_Contract)	The total number of transactions, including transactions for creating contracts.
Total_Ether_Sent	Total ether sent from the account address.
Total_Ether_Received	Total Ether received at the account address.
Total_Ether_Sent_Contracts	Total ether sent to contract addresses.
Total_Ether_Balance	Total Ether balance after all transactions.
Total_ERC20_Tnxs	Total number of ERC20 token transfer transactions.
ERC20_Total_Ether_Received	Total transactions received for ERC20 token in ether.
ERC20_Total_Ether_Sent	Total transactions sent for ERC20 token in ether.
ERC20_Total_Ether_Sent_Contract	The entire ERC20 token that was sent across the ether to other contracts.
ERC20_Uniq_Sent_Addr	The quantity of transactions with ERC20 tokens transferred to distinct account addresses.
ERC20_Uniq_Rec_Addr	The quantity of transactions with ERC20 tokens transferred to distinct account addresses.
ERC20_Uniq_Rec_Contract_Addr	The quantity of transactions in ERC20 tokens that came from distinct contract addresses.

ERC20_Avg_Time_Between_Sent_Tnx	The mean number of minutes between ERC20 token transactions that are sent.
ERC20_Avg_Time_Between_Rec_Tnx	The average number of minutes it takes to receive an ERC20 token for a transaction.
ERC20_Avg_Time_Between_Contract_Tnx	The typical interval in seconds between an ERC20 token's transaction transmission to contracts.
ERC20_Min_Val_Rec	The smallest amount of Ether that the account has received from ERC20 token transactions.
ERC20_Max_Val_Rec	The highest amount of Ether that was received from ERC20 token transactions on the account.
ERC20_Avg_Val_Rec	Average amount of Ether received in exchange for ERC20 tokens on the account.
ERC20_Min_Val_Sent	The smallest amount of Ether that is transferred to the account from ERC20 token transactions.
ERC20_Max_Val_Sent	The highest amount of ether sent from ERC20 token transactions to the account.
ERC20_Avg_Val_Sent	The average value of ether sent from ERC20 token transactions to the account.
ERC20_Uniq_Sent_Token_Name	Number of ERC20 tokens transferred.
ERC20_Uniq_Rec_Token_Name	Number of ERC20 tokens received.
ERC20_Most_Sent_Token_Type	The most sent token to the account via an ERC20 transaction.
ERC20_Most_Rec_Token_Type	The token most received by the account via ERC20 transactions.

4.2. Preprocessing

The pre-processing stage is considered one of the basic and crucial stages in the field of machine learning. At this stage, the data is prepared through a set of steps mentioned below to make the data interpretable and analyzed by artificial intelligence models, which contributes to making these models capable of learning, perception, understanding, and then prediction and decision-making. The following section explains the basic steps in the pre-processing stage.

4.2.1. Processing Missing Values

This step involves removing impurities and noise from the data and making it interpretable. During this stage, unread values (or missing values) are removed. Several mathematical or statistical methods are used to eliminate missing values. For example, explicit values (integers or decimals) can be used to fill in missing values, or calculate the arithmetic mean, standard deviation etc., for the set of other features and replace them with the missing values. This step contributes to getting rid of outliers that negatively reflect on the learning process of machine learning models of all types.

4.2.2. Dataset Splitting

Artificial intelligence models mimic the learning concept of the human mind, which works on the principle of learning, understanding, perception, then prediction and decision-making. This requires dividing the data into two groups. First, which has a larger size, targets the principle of learning and training in artificial intelligence models. While the remaining group (small in size) works to test the model and evaluate its performance. In the proposed model, the data set was divided by 80% into a subset to train the model, and the remaining 20% to test and evaluate its effectiveness [21].

4.2.3. Normalization

The concept of normalizing numerical values involves converting them into a linear vector without affecting the actual value to facilitate the training process for artificial intelligence models. There are several common ways to obtain a vector from training data. In the proposed model, the Min-Max normalization principle was used to transform a data set within a specific range, as the mathematical equation explains [22].

Where

$$f_{new} = \frac{f - \min(f)}{\max(f) - \min(f)}$$

f : is the feature value.

f_{new} : is the normalized f .

$\min(f)$: is the minimum amount for a feature f .

$\max(f)$: is the maximum amount for a feature f .

4.2.4. Data Balance

The Ethereum transaction fraud detection dataset as shown in Figures 5 and 6, suffers from class imbalance, which results in machine learning models leaning toward the most popular class and making unreliable predictions during evaluations.

```
0    7662
1    2179
Name: count, dtype: int64
```

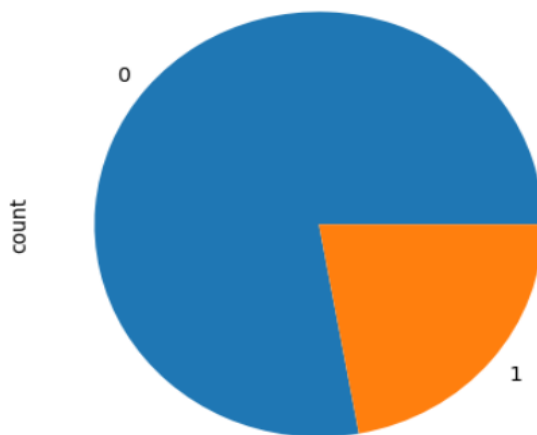


Figure 5. Distribution classes in dataset

The approach proposes to balance the dataset by applying the RandomOverSampler technique, which samples the rare class and generates new samples to balance the data.

This contributes to improving the reliability of the model's predictions and increasing its efficiency in dealing with diverse and imbalanced data [23]. Figure7 shows the balancing classes in dataset.

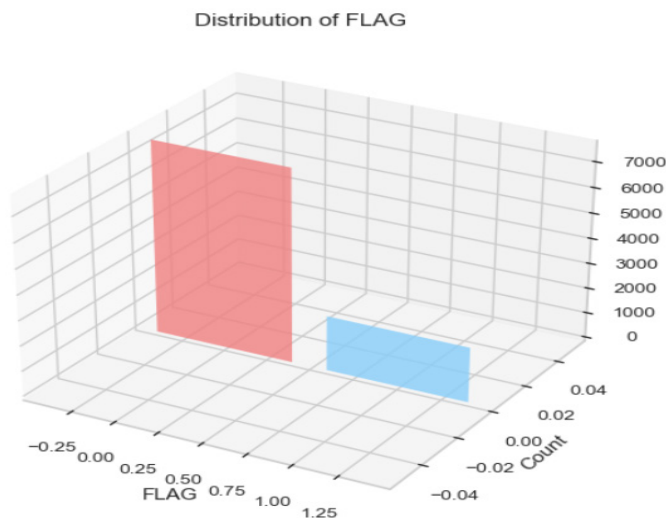


Figure 6. Unbalancing classes in dataset

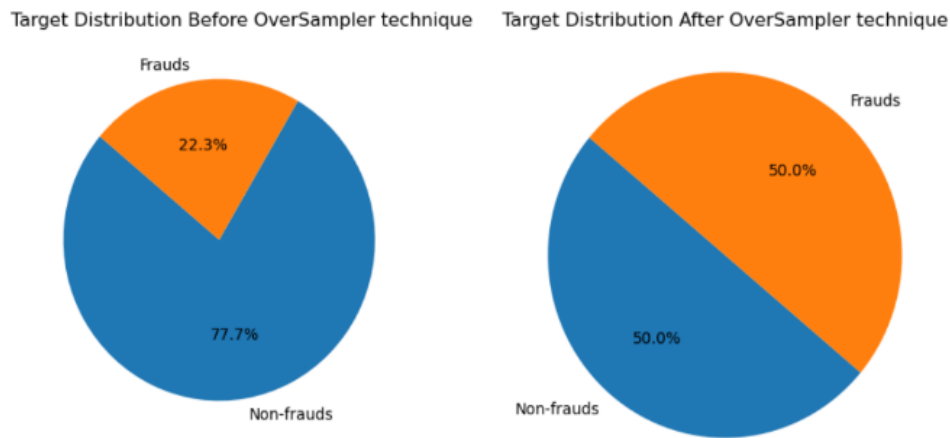


Figure 7. Balancing dataset with Random Over Sampler technique.

4.3. Machine Learning

There are studies that have investigated the possibility of using machine learning techniques to analyze intrusion detection in Internet networks [24]. Efforts have focused on updating existing machine learning models to identify these attacks, whether by improving existing models or developing new models from scratch. These efforts include determining verification threshold values to determine whether a network access request is legitimate or an attack [25], [26].

4.3.1. PyCaret Machine Learning Library

The PyCaret library is a machine learning library and low-code framework for classification and prediction purposes. It uses multiple frameworks such as Scikit-learn, and is able to provide interactive dashboards to help users understand the analysis and results more clearly. It was released as an open source library in April 2020. PyCaret relies on a small amount of code to run machine learning models, which can then be easily deployed in networks vulnerable to cyberattacks. Released as an open-source library in April 2020, PyCaret features the ability to run machine learning models with a small amount of code, making it easy to deploy in vulnerable networks. The model includes twelve machine learning classifiers trained on two training and test datasets [27].

4.3.2. The Python Package

Technology is influenced by many different types of innovations, making Python crucial in supporting innovation in all fields [28].

The Python environment is widely considered an excellent choice for implementing machine learning scenarios, thanks to the abundance of powerful packages and libraries related to this topic [29]. The

Python language provides many libraries that facilitate data analysis and adaptation to work with artificial intelligence algorithms (machine learning and deep learning).

4.3.2.1. The Scientific Term (NumPy)

The NumPy library deals with multi-dimensional digital data such as arrays and digital strings (integer and decimal) and provides many processors in the configuration convenience [30].

4.3.2.2. Keras Library

Python programming provides an open source neural library known as Keras. Keras helps run many programs such as Theano, MXNet, TensorFlow, in addition to CNTK deep learning. Keras contributes to rapid experimentation with deep neural networks in a way that makes it a scalable model [31].

4.3.2.3. Pandas

The Pandas library is also considered an open-source neural library provided by Python programming, as it helps developers deal with real-world data easily by reading it in a special data frame and then configuring it in the pre-processing stage and preparing it to work with artificial intelligence models [32].

4.3.2.4. Matplotlib

Matplotlib Analyzing and understanding the behavior of real-world data makes an important contribution to defining what automated work should look like. For example, when analyzing data, the number of categories that make up the target data, their size etc., are known. Matplotlib helps developers analyze data and transform it into graphical plots that are easy to understand and interpret in the pre-processing stage [33].

4.3.2.5. Performance Evaluation Metrics

The stage of testing and evaluating the performance of AI models represents one of the most important steps and may be considered the decisive step that determines the success or failure of the model. It also contributes to making decisions that improve the performance or prepare the data differently in the pre-processing stage. There are several metrics provided by the Python environment to evaluate the model's performance. The sklearn metrics library provided in the Scikit-learn package provides many performance metrics such as accuracy, recall, precision, and recall, as shown in equations (1-4). In addition to the confusion matrix shown in Figure 8, which shows the relationship between real-world values and positive and negative expectations [34], [14], [20]:

$$\text{accuracy} = \frac{TP+TN}{TP+TN+FN+FP} \quad (1)$$

$$\text{precision} = \frac{TP}{TP+FP} \quad (2)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (3)$$

$$\text{F1 - score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4)$$

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Figure 8. The confusion matrix

5. Results and Discussion

In this section, the results of applying the open source machine learning library (Pycarte) to perform the research task of analyzing and detecting anomalous and fraudulent behavior of digital currencies in the Ethereum network are presented, in addition to using machine learning techniques. The proposed system is then tested on the Ethereum transactions dataset. Finally, classification metrics based on confusion matrix distributions are applied.

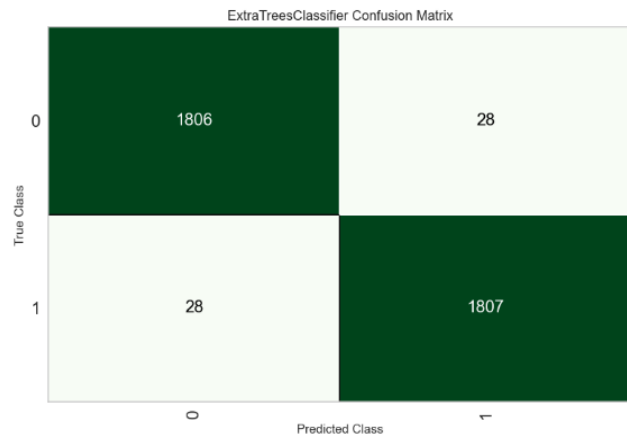


Figure 9. Confusion matrix for pycart model

Table 2. Classifier report for extratree model

Models	Accuracy	AUC	Recall	Prec.	F1-score
et	0.9862	0.9989	0.99	0.9827	0.9863
lightgbm	0.9853	0.9989	0.9867	0.984	0.9853
rf	0.9814	0.9984	0.9839	0.9791	0.9815
knn	0.9702	0.9908	0.9857	0.9561	0.9707
gbc	0.9672	0.9964	0.9717	0.963	0.9673
dt	0.9636	0.9636	0.9666	0.9608	0.9637
ada	0.9522	0.9909	0.9561	0.9489	0.9524
qda	0.9176	0.9654	0.9386	0.901	0.9193
lr	0.8935	0.952	0.9068	0.8834	0.8948
svm	0.8908	0.9426	0.9033	0.8814	0.892
ridge	0.8891	0.9492	0.9112	0.8728	0.8915
lda	0.889	0.9491	0.9112	0.8726	0.8914
nb	0.7743	0.8962	0.9178	0.7134	0.8027

The next section shows the confusion matrix for all algorithms in the Pycart model.

Confusion Matrix for Light Gradient Boosting Machine

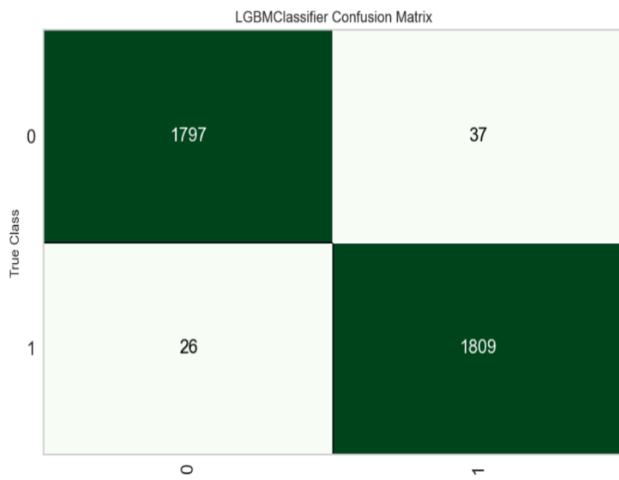


Figure 10. Confusion matrix for LightGBM model

Confusion Matrix for Random Forest Classifier

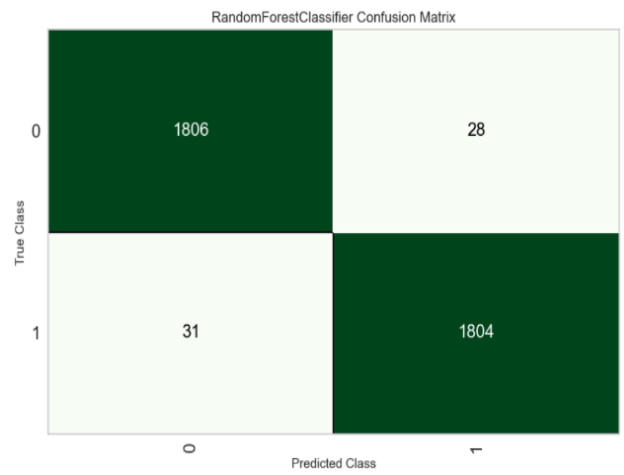


Figure 11. Confusion matrix for Random Forest mode

Confusion Matrix for K Neighbors Classifier

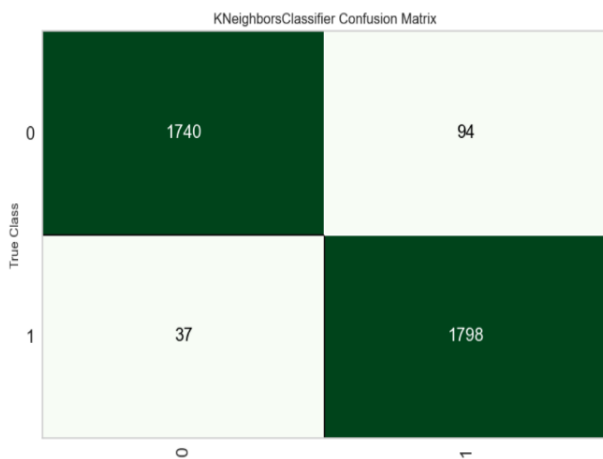


Figure 12. Confusion matrix for KNN model

Confusion Matrix for Gradient Boosting Classifier

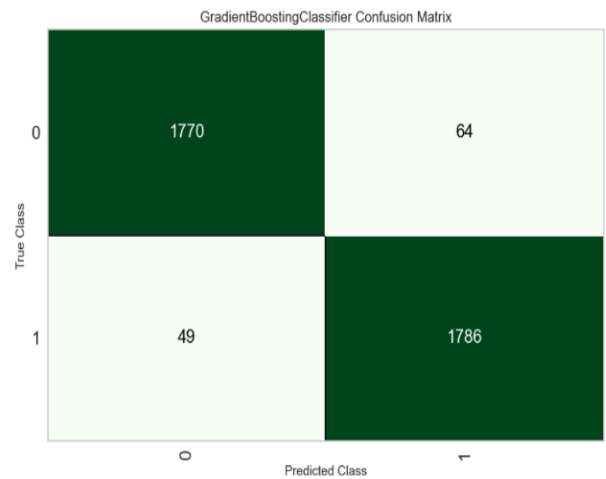


Figure 13. Confusion matrix for GBC model

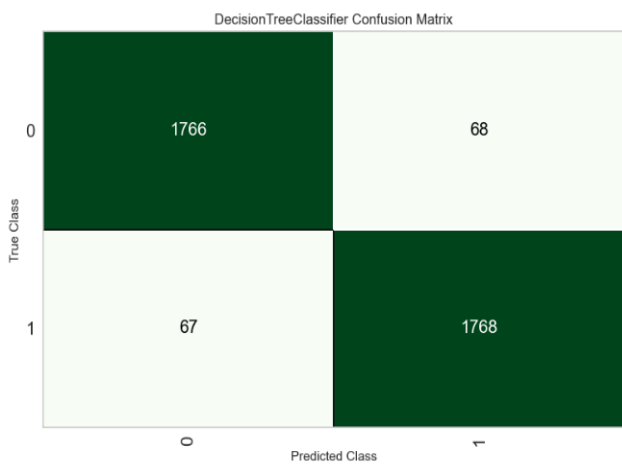


Figure 14. Confusion matrix for DT model

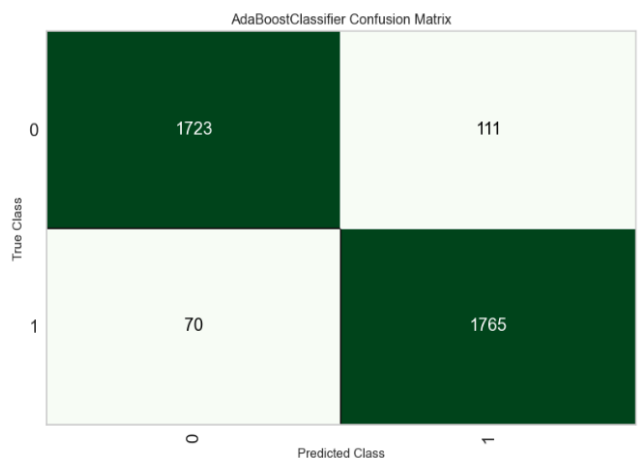


Figure 15. Confusion matrix for ada model

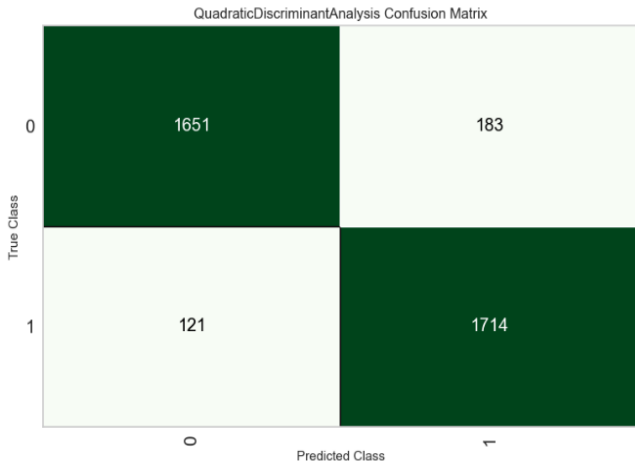


Figure 16. Confusion matrix for qda model

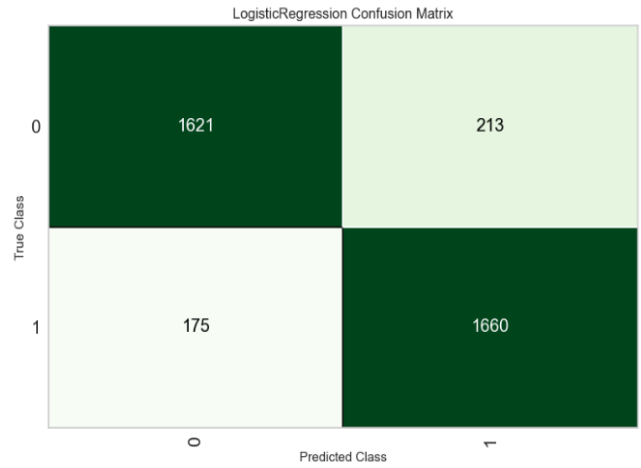


Figure 17. Confusion matrix for LR model

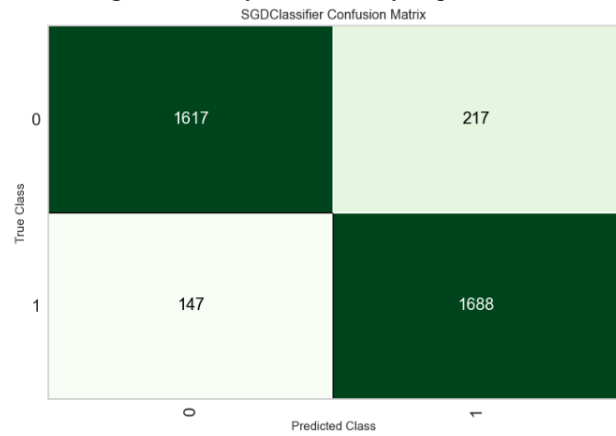


Figure 18. Confusion matrix for SVM - Linear Kernel

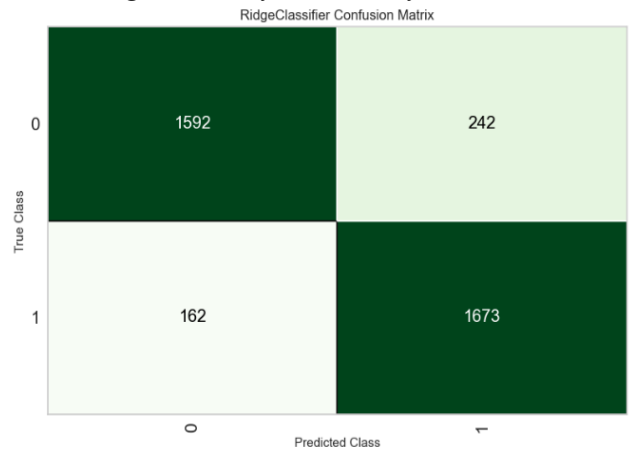


Figure 19. Confusion matrix for Ridge Classifier

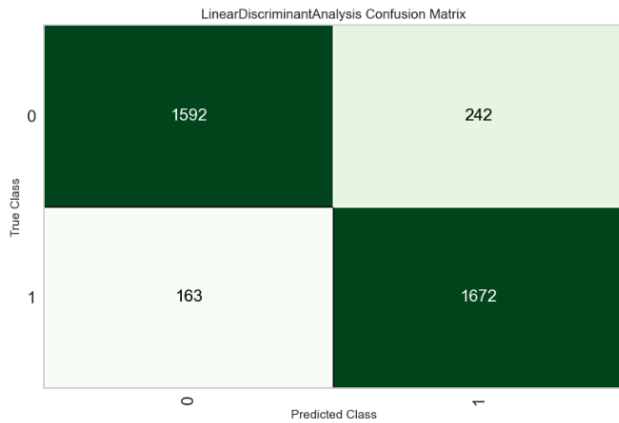


Figure 20. Confusion matrix for Linear Discriminant analysis

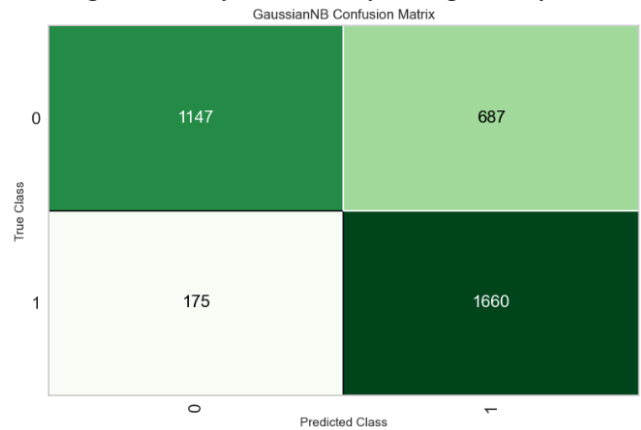


Figure 21. Confusion matrix for Naive Bayes

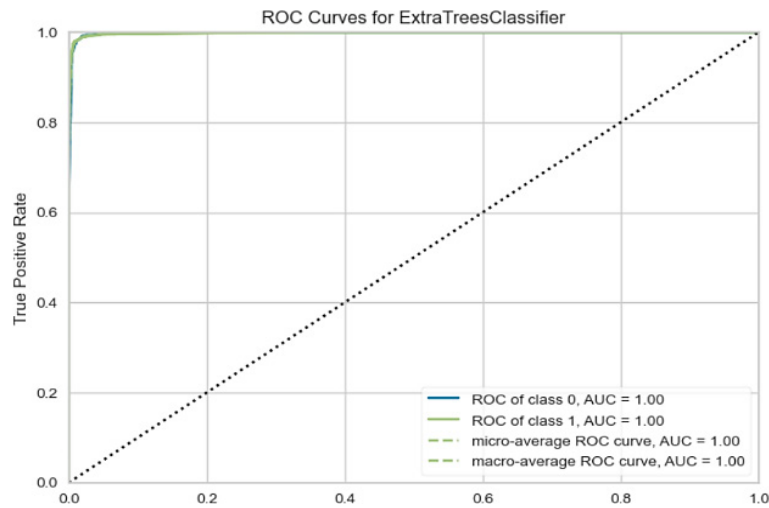


Figure 22. ROC curve for best model

In this section, when looking at the predictive results for the classification task achieved from the performance of the model proposed in this paper and presented in Table 2, it is noticeable that there is a discrepancy in the performance of the 12 classifiers that were measured based on the measures of accuracy, recall, precision, and F1 score. In addition, the measure of the area under the receiver operating characteristic curve (AUC) was adopted. Extra Trees (ET) and LightGBM achieved the highest accuracy scores between 98.62% and 98.53%, followed by the Random Forest (RF) classifier with a predictive accuracy of 98.14%. When applying the evaluation measure related to the area under the curve (AUC), the superiority of all classifiers is noticed, which contributes to the reliability of the predictions of the proposed model. The Extra Trees model also achieved the highest F1 score of 98.63% compared to the remaining 11 classifiers.

In general, these results indicate a successful comparison of the performance of the proposed model. This contributes to making critical decisions and future recommendations regarding protecting electronic currency trading markets that operate on the principle of data and distributed systems from the risk of abnormal financial behavior of users and providing a safe environment free of financial fraud of all kinds.

6. Conclusion

The PyCaret machine learning library is used for analysis and discovery, and the results show that it is an effective tool for demonstrating the effectiveness of analysis and discovery. An example of anomalous

and fraudulent behavior was found within the Ethereum network. Next, to achieve the main goal of the research, a combination of machine learning methods and classification metrics analysis to explore the effectiveness of different algorithms in identifying fraudulent transactions was chosen.

The proposed model can be considered a reliable candidate for future research work in the field of financial analysis in networks operating on the principle of databases, distributed and decentralized systems where the 12 works were recorded.

Bickert has high accuracy in the performance evaluation stage, which contributes to providing reliable security measures within the blockchain and Ethereum networks and detecting abnormal beneficiary behavior in electronic exchanges. The results of the performance evaluation of the machine learning models showed clear differences in their behavior with the experimental data set. Some classifiers, such as the Random Tree Forest, the Additive Tree Classifier, and the Visual Gradient Boosting Machine, showed the highest degrees of accuracy, with percentages ranging from 0.9814 to 0.9862. These models also showed high areas under the curve (AUC), indicating strong discriminatory power. It is worth noting that K Neighbors Classifier and Gradient Boosting Classifier also performed well, achieving accuracies above 0.96 and AUC scores above 0.99. However, some models, such as Naive Bayes, showed lower accuracy and AUC scores, indicating limitations in their ability to effectively identify fraudulent transactions. Overall, these results underscore the importance of choosing appropriate machine learning models for fraud detection tasks.

References:

- [1].Aziz, R. M., Baluch, M. F., Patel, S., & Ganie, A. H. (2022). LGBM: a machine learning approach for Ethereum fraud detection. *International Journal of Information Technology*, 14(7), 3321-3331. Doi:10.1007/s41870-022-00864-6.
- [2].Bartoletti, M., Carta, S., Cimoli, T., & Saia, R. (2020). Dissecting Ponzi schemes on Ethereum: identification, analysis, and impact. *Future Generation Computer Systems*, 102, 259-277.
- [3].Hu, T., Liu, X., et al. (2021). Transaction-based classification and detection approach for Ethereum smart contract. *Information Processing & Management*, 58(2), 102462.
- [4].Yuan, Q., et al. (2020). Detecting phishing scams on ethereum based on transaction records. *2020 IEEE international symposium on circuits and systems (ISCAS)*, 1-5. IEEE.
- [5].Rababaah, H., & Hakimzadeh, D. H. (2005). Distributed Databases fundamentals and research. *Advanced Database B*, 561.
- [6].Zheng, Z., Xie, S., Dai, H. N., Chen, X., & Wang, H. (2018). Blockchain challenges and opportunities: A survey. *International journal of web and grid services*, 14(4), 352-375.
- [7].Zheng, P., Zheng, Z., Wu, J., & Dai, H. N. (2020). Xblock-eth: Extracting and exploring blockchain data from ethereum. *IEEE Open Journal of the Computer Society*, 1, 95-106.
- [8].Howell, S. T., Niessner, M., & Yermack, D. (2020). Initial coin offerings: Financing growth with cryptocurrency token sales. *The Review of Financial Studies*, 33(9), 3925-3974.
- [9].Li, X., Jiang, P., Chen, T., Luo, X., & Wen, Q. (2020). A survey on the security of blockchain systems. *Future generation computer systems*, 107, 841-853.
- [10]. Einstein, A., Podolsky, B., & Rosen, N. (1935). Can quantum-mechanical description of physical reality be considered complete?. *Physical review*, 47(10), 777.
- [11]. Aziz, R. M., Baluch, M. F., Patel, S., & Kumar, P. (2022). A machine learning based approach to detect the Ethereum fraud transactions with limited attributes. *Karbala International Journal of Modern Science*, 8(2), 139-151. Doi: 10.33640/2405-609X.3229.
- [12]. Hu, T., et al. (2021). Transaction-based classification and detection approach for Ethereum smart contract. *Information Processing & Management*, 58(2), 102462. Doi: 10.1016/j.ipm.2020.102462.
- [13]. Kabla, A. H. H., Anbar, M., Manickam, S., & Karupayah, S. (2022). Eth-PSD: A machine learning-based phishing scam detection approach in ethereum. *IEEE Access*, 10, 118043-118057. Doi: 10.1109/ACCESS.2022.3220780.
- [14]. Qin, J., et al. (2022). Research and application of machine learning for additive manufacturing. *Additive Manufacturing*, 52, 102691.
- [15]. Sallam, A., et al. (2022). Fraudulent account detection in the Ethereum's network using various machine learning techniques. *International Journal of Software Engineering and Computer Systems*, 8(2), 43-50. Doi:10.15282/ijsecs.8.2.2022.5.0102.
- [16]. Kılıc, B., Sen, A., & Özturan, C. (2022, September). Fraud detection in blockchains using machine learning. In *2022 Fourth International Conference on Blockchain Computing and Applications (BCCA)*, 214-218. IEEE. Doi: 10.1109/BCCA55292.2022.9922045.
- [17]. Pranto, T. H., et al. (2022). Blockchain and machine learning for fraud detection: A privacy-preserving and adaptive incentive based approach. *IEEE Access*, 10, 87115-87134. Doi: 10.1109/ACCESS.2022.3198956.
- [18]. Tripathy, N., Balabantaray, S. K., Parida, S., & Nayak, S. K. (2024). Cryptocurrency fraud detection through classification techniques. *International Journal of Electrical and Computer Engineering (IJECE)*, 14(3), 2918-2926. Doi: 10.11591/ijece.v14i3.pp2918-2926.
- [19]. Taher, S. S., Ameen, S. Y., & Ahmed, J. A. (2024). Advanced Fraud Detection in Blockchain Transactions: An Ensemble Learning and Explainable AI Approach. *Engineering, Technology & Applied Science Research*, 14(1), 12822-12830. Doi: 10.48084/etasr.6641.
- [20]. Scikit-learn. (2023). *Scikit-learn: Machine Learning in Python*. Scikit-learn. Retrieved from: <https://scikit-learn.org/stable/index.html> [accessed: 10 April 2024]
- [21]. Roy, R. (2021). *Ethereum Fraud Detection*. Geek Culture. Retrieved from: <https://bobrupakroy.medium.com/ethereum-fraud-detection-ba4e1d8b262a> [accessed: 15 April 2024].
- [22]. Liao, S., Jain, A. K., & Li, S. Z. (2015). A fast and accurate unconstrained face detector. *IEEE transactions on pattern analysis and machine intelligence*, 38(2), 211-223.
- [23]. Kanezashi, H., Suzumura, T., Liu, X., & Hirofuchi, T. (2022). Ethereum fraud detection with heterogeneous graph neural networks. *arXiv preprint arXiv:2203.12363*.
- [24]. Torres-García, A. A., et al. (2022). Pre-processing and feature extraction. In *Biosignal processing and classification using computational learning and intelligence*, 59-91. Academic Press. Doi: 10.1016/B978-0-12-820125-1.00014.
- [25]. Zhu, X., Lei, Z., Liu, X., Shi, H., & Li, S. Z. (2016). Face alignment across large poses: A 3d solution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 146-155.
- [26]. Haq, S., & Singh, Y. (2018). Botnet detection using machine learning. In *2018 Fifth International Conference on Parallel, Distributed and Grid Computing (PDGC)*, 240-245. IEEE.
- [27]. Stevanovic, M., & Pedersen, J. M. (2014, February). An efficient flow-based botnet detection using supervised machine learning. In *2014 international conference on computing, networking and communications (ICNC)*, 797-801. IEEE. Doi: 10.1109/ICCNC.2014.6785439.

- [28]. Iqbal, F. B., Biswas, S., & Urba, R. (2021). *Performance analysis of intrusion detection systems using the PyCaret machine learning library on the UNSW-NB15 dataset*. [Doctoral dissertation, Brac University].
- [29]. Sharma, A., Khan, F., Sharma, D., Gupta, S., & Student, F. Y. (2020). Python: the programming language of future. *Int. J. Innovative Res. Technol*, 6(2), 115-118.
- [30]. Jalolov, T. S. (2023). Python instrumentlari bilan katta ma'lumotlarni qayta ishlash. *Educational Research in Universal Sciences*, 2(11 special), 320-322.
- [31]. Harris, C. R., *et al.* (2020). Array programming with NumPy. *Nature*, 585(7825), 357-362.
- [32]. Dürr, O., Sick, B., & Murina, E. (2020). *Probabilistic deep learning: With python, keras and tensorflow probability*. Manning Publications.
- [33]. Saabith, A. S., Vinothraj, T., & Fareez, M. (2020). Popular python libraries and their application domains. *International Journal of Advance Engineering and Research Development*, 7(11), 18-26.
- [34]. Sial, A. H., Rashdi, S. Y. S., & Khan, A. H. (2021). Comparative analysis of data visualization libraries Matplotlib and Seaborn in Python. *International Journal*, 10(1), 277-281