International Journal of Industrial Engineering
2026;10(2):20-29.
ISSN: 2456-8449

K-Means Clustering for Identifying Suspicious Patterns in Ethereum Networks

Uzzma Khan[1], Rahul Vijayalakshmi[2*]

[1]*Ontario Tech University, Queens Quay W, Toronto, Canada.*
[2]*Department of Applied Cyber Security, Technological University Dublin, Dublin, Ireland.*

[*]*Corresponding author: careeratrahulav@gmail.com*

Abstract. The use of Ethereum cryptocurrency has been increasing recently, and thus there has been a growing demand for detecting fraudulent activities to protect digital financial systems. This paper seeks to apply the K-Means clustering algorithm in detecting fraud among the Ethereum transactions based on account transactions and behavior characteristics. The Ethereum Fraud Detection Dataset from Kaggle will be used to analyze fraudulent and genuine transactions. The dataset includes account addresses, number of transactions per account, transaction intervals, amount sent/ received from the account in terms of Ether, account balances, distinct sender/receiver addresses, contract calls and ERC20 Token transactions characteristics. The data pre-processing, feature selection, feature scaling and 80:20 train-test splitting will be part of the detection process. The results obtained from experimentation show that the model has an accuracy of 98.07% and F1-score of 98.08%. This confirms the ability of the method to differentiate between fraudulent transactions and genuine activities. The method allows for systematic analysis of cryptocurrency transactions and the detection of abnormal behaviors. Further verification through experimentation is needed especially due to the skewed classes in the dataset.

Keywords: Blockchain, Ponzi Schemes, Ethereum and Smart Contracts.

INTRODUCTION

In the time since its launch, Ethereum has grown into a leading blockchain platform. Every day, miners create over 6,000 blocks, and the number of Ethereum accounts has surpassed 207 million. Because of its popularity, Ethereum is also a target of fraud, therefore identifying and stopping these scams is essential to the network's long-term viability. The primary objective of this work is to examine and contrast several approaches to fraud detection, with a secondary goal of reducing the number of false positives. Based on these comparisons, the authors will then propose a model that they believe will be the most effective in detecting fraud in Ethereum transactions. The use of blockchain technology is growing rapidly, and its influence is spreading beyond the realms of digital assets and finance. The availability of a decentralized public ledger makes it simple to verify the authenticity of blockchain-based transactions and accounts. However, some bad actors try to defraud bitcoin holders, which damages blockchain trust. The purpose of this research is to find suspicious activity in the two most popular cryptocurrency transaction networks, Ethereum and Bitcoin, to uncover fake accounts and transactions. Utilizing GPU-accelerated machine learning models, including as Support Vector Machines, Random Forest, and Logistic Regression, we extract details from more than 500,000 Ethereum accounts' verified transactions and 30 million Bitcoin transactions.

Through sensitivity analysis, we provide light on the significance of features and train reliable models for use in automated fraud detection systems. On the Bitcoin dataset, the trained models get a recall and accuracy of 0.987, while on the Ethereum dataset, they obtain 80.2% and 0.835, respectively. This paper's anomaly detection investigation in the Bitcoin blockchain has broad applicability to other blockchain networks, such as those in the public sector, financial intelligence, health service, and other related industries. A growing number of people are interested in blockchain and cryptocurrencies with the introduction of technology 4.0. Since the Ethereum network is now a popular platform for developers to build smart contracts and release decentralized applications, it is crucial to include fraud detection methods over it. The use of machine learning techniques, especially neural networks, to identify fraud has been expanding over the years, with encouraging results. We thus provide a neural network-based method for Ethereum fraud detection in this research and compare it to its contemporaries to verify the impact of its performance. The neural network outperforms the other models, including Logistic Regression, Support Vector Machines, Gaussian Naive Bayes, and K-nearest neighbor, with an accuracy of around 97.09%.

Received: 17.04.2026 Revised: 08.06.2026 Accepted: 24.06.2026
Licensed under a CC-BY 4.0 license | Copyright (c) by the authors

It follows that neural networks may learn complicated dataset patterns and, as a consequence, determine if a transaction is legitimate or fraudulent. The development of reliable and secure systems for detecting fraud on Ethereum and other blockchain platforms is therefore aided by our study.

Ethereum smart contracts, a household brand in the cryptocurrency industry and a novel approach to financial transactions, have attracted a great deal of academic interest. Their usefulness is such that we can do away with a separate third-party library that would otherwise be required to provide unidentified parties access to computerized contract data. We can all see that e-commerce is booming and will keep on increasing, but that doesn't mean it's completely devoid of frauds and unethical practices. To accurately identify all of these fraudulent and immoral transactions, a deep learning model was used in this work. Additionally, metaheuristic optimizations were used in this work to further improve the model. The purpose of this program is to identify fraudulent transactions using Cuckoo Search (CS), and it uses a Genetic program to optimize the exploratory phase. Every vulnerability in the CS approach is addressed by the algorithm. Light Gradient Boosting Machine, Support Vector Classification, Multi-Layer Perceptron, XG-Boost, Logistic Regression, and Random Forest were among the techniques that were contrasted with our suggested model to establish solid research grounds in our article. With a 98.6 percent accuracy rate, the suggested model surpasses other notable models such as SVC, KNN, LGBM Classifier, RF, and beyond. Ethereum is a decentralized application platform that uses blockchain technology to distribute smart contract versions to thousands of users throughout the world. If you're looking for a worldwide currency that can be used to trade value without any outside interference or oversight, go no further than Ethereum. With the expansion of online trade comes the greatest danger to trade security: the rise of corrupt practices such as bribery, phishing, and money laundering. Threats like fraud and cyberattacks that are intrinsic to blockchain technology emphasize the necessity of robust defenses. Maintaining trust in the financial system and keeping it stable requires an accessible network that is resistant to these types of attacks.

In order to identify instances of Ethereum fraud, this study employs a variety of classification techniques, such as Logistic Regression (LR), K-Nearest Neighbors (KNN), Random Forest (RF), and Extreme Gradient Boosting (XGBoost). We are making use of a dataset that contains both legitimate and known fraudulent transactions using the Ethereum cryptocurrency. The "XGBoost" model stands out because it can spot variations that might cause trouble and alert you to them. The safety and reliability of the Ethereum ecosystem is enhanced by supervised learning's ability to identify fraud. There has to be more regulation on Ethereum because of all the fraudulent activity there. This is stopping the blockchain ecosystem from growing properly. Research into data mining for fraud detection faces substantial obstacles in the Ethereum transaction environment due to various mismatches in the frequency and kinds of account interactions. We provide a dual self-supervision improved Ethereum fraud detection system, Meta-IFD, based on the idea of meta-interactions, which we suggest as a means to optimize interaction behaviors in Ethereum. This framework begins by introducing a generative self-supervision mechanism to enhance account interaction features. It then moves on to a contrastive self-supervision mechanism to distinguish between different patterns of behavior. Finally, it uses multi-view interaction feature learning to characterize account behavioral representations and mine for potential fraud risks. Extensive testing on actual Ethereum datasets proves that our approach is the best at identifying typical Ponzi schemes and phishing scams that occur on Ethereum. To top it all off, the generative module can fix Ethereum's interaction distribution imbalance, and the contrastive module makes the framework far better at detecting patterns of behavior. Go to https://github.com/GISec-Team/Meta-IFD to get the source code. A review of the relevant literature will follow in Section 2, and the ARIMA Model -based Crude oil price forecasting will be covered in Section 3. Section 4 then discusses the outcomes and feedback on the provided dataset in an effort to establish a mechanism in crude oil price forecasting. Finally, covered Systems' long-term objectives and the system's overall performance.

LITERATURE SURVEY

People are more vulnerable to scams because, with the rise of e-commerce, more and more transactions are taking place online. As a result, banks are under constant pressure to innovate and modernize their business models. Credit card fraud was detected using Machine Learning methods; nevertheless, dealing with real-time data poses challenges for Machine Learning [1]. Therefore, combining blockchain technology with machine learning may enhance the model's efficacy and precision. The suggested technique employs machine learning methods to safeguard the Ethereum dataset by checking for fraudulent transactions. For the given dataset, XGBoost achieved the best accuracy of all the classifiers at 99.21%. The banking industry is seeing a corresponding increase in fraud activities due to the advent of crypto-currencies like Ethereum and Bitcoin [2-3]. Because cryptocurrencies are decentralized, anonymous, and not controlled by a central authority, it becomes

incredibly difficult to identify fraudulent transactions using them. In order to detect bogus Ethereum addresses, we used a decision-tree based machine learning model with a boosting technique, namely XGBoost, in this article. First, we used a cross-validation technique to rank the four best decision tree learning approaches—CART, random forest, LGBM, and XGBoost—on our fraud dataset. Then, we choose the best one based on accuracy. Finally, after fine-tuning the XGBoost model with the optimal hyper parameters, we chose it above CART, random forest, and gradient boosting tree. By using only 80% of the training data, we were able to train the model to achieve an accuracy of over 96% on test data. In addition, the model's efficiency in a real-world setting has been shown by considerable experimental study. One kind of cryptocurrency, Ethereum, was the focus of this research, which looked at how machine learning models may identify fraudulent transactions [4]. There were 9840 transactions in the dataset, representing both valid and fraudulent ones. The parameters included average, minimum, and maximum amounts received and sent, the number of unique addresses used for sending and receiving ether, the time duration between transactions, and the remaining balance following the transactions. The amount of the transaction falls anywhere between 12000.0 and 52000.0. In order to find the best machine learning model for detecting fraudulent transactions, we tested four of them: Decision Tree, Random Forest Classifier, Catboost Classifier, and XGBoost Classifier [5]. With near-perfect accuracy (99.65%) and strong precision, recall, and F1-scores, XGBoost has emerged as the most promising model. Notably, XGBoost's strong performance implies it has the ability to identify fraudulent Ethereum transactions with high accuracy. On the other hand, models such as Random Forest and Decision Tree perform quite poorly.

By developing a practical and scalable fraud-detecting tool, this work highlights the importance of machine learning in improving Ethereum transaction fraud detection and advancing blockchain cybersecurity [6-7]. Ethereum is the first platform that allows the new Decentralized Applications (DApps) to operate on top of the blockchain network, which is one of the many reasons why it draws more investors, academics, and even fraudsters. Security concerns have a natural tendency to accompany DApps' complex semantics and applications; these concerns have recently attracted a lot of attention from both academia and business as a result of the damage they have done to Dapps [8]. Studies on the usefulness of Intrusion Detection Systems in identifying assaults based on Ethereum are, hence, critically necessary. Therefore, this study is one of the first reviews to examine the use of intrusion detection systems (IDS) in identifying attacks that target Ethereum. Also included in this article are a comprehensive list of all possible attacks against Ethereum, along with the vulnerabilities that might lead to them and the repercussions of each assault [9-10]. Furthermore, since the Ethereum platform's inception in 2015, this study examines all research pertaining to the detection of Ethereum attacks that are based on intrusion detection systems. At last, this study delves into the unanswered questions surrounding vulnerabilities and assaults, obstacles, and potential future paths. Blockchain technology is the backbone of digital currency systems such as Ethereum and Bitcoin.

One goal of blockchain technology is to do away with intermediaries like banks [11]. The immutability and transparency of this technology have led to its expansion beyond cryptocurrencies in recent years, with applications in areas such as healthcare, education, banking, energy, government, and the Internet of Things (IoT), all of which have benefited from increased privacy, quicker transactions, and greater security. Here, we looked at Ethereum blockchain-based illicit accounts and offered a model to identify fraud using a combination of decision tree (j48), Random Forest, and K-nearest neighbors (KNN) [12-13]. A data set with 42 features was applied to the dataset received from Kaggle.com using these techniques. A fresh dataset is constructed with only six characteristics, utilizing the correlation coefficient to determine which ones were most useful. According to our findings, all three algorithms significantly enhance time measures, whereas the Random Forest approach particularly improves the F measure. As we have seen in the last few years, blockchain is here to stay, and crypto crimes reached a high [14]. The manual investigation of crimes will also become very difficult in the future due to the anticipated increase in blockchain transaction throughputs. In order to keep tabs on blockchains and identify any fraudulent activity, we want efficient and automated methods.

We provide a method to identify the Ethereum blockchain's banned addresses in this article. We began by amassing a database of all banned addresses and Ethereum blockchain transaction data. After that, we built the Ethereum transaction graph and retrieved address characteristics, including some global statistics like pagerank [15]. Lastly, we anticipated the address class by training the models of common machine learning methods. Our findings demonstrate that we can anticipate banned addresses with an accuracy of over 97%. The decentralization, cryptographic verification, and immutability of blockchain technology have made it a popular choice for many different types of businesses. Nevertheless, blockchain networks are still vulnerable to cyberattacks, despite their great security features. Recently, the blockchain system has been under assault from fraudulent actors,

necessitating immediate action [16-17]. More and more, these problems are being solved by using machine learning and deep learning models. A big complaint, however, is that these models aren't transparent enough because of their black box design. An method to improving Ethereum's fraud detection systems is presented in [18]. This paper introduces IFS-TABPFN, an open-source fraud detection system for Ethereum that is both effective and transparent. In order to construct five neural network-based deep learning models, an interpretable feature selection method using Shap values and optimized gradient boosting was implemented. The following models were included in the set: Multilayer Perceptron's (MLP), Long Short-Term Memory (LSTM), Convolutional Neural Networks (CNN), Convolutional Neural Networks with Long Short-Term Memory (CLSTM), and Tabular Prior-Data Fitted Network (TabPFN) [19]. The findings show that IFS-TABPFN outperforms other neural networks and current systems, reaching 99.2% accuracy in a matter of seconds.

PROPOSED SYSTEM

To begin with, we want an Ethereum training dataset that includes both Ponzi and non-Ponzi smart contracts. Data collecting is sometimes the most time-consuming part of fraud detection initiatives since it requires extensive human verification and study to differentiate between fraudulent and legitimate cases. The availability of an open-source dataset with the addresses of 184 Ponzi schemes is due to. Only 172 of them had their bytecode recoverable; the other 8 had self-destruct code. We take these contracts at face value since we have personally checked each address to make sure it is not a Ponzi scam. On August 7, 2015, the first Ponzi smart contract was published to the Ethereum blockchain; on August 27, 2017, the most recent one was released.

To match the time of a Ponzi scam, we used a web scraper to gather 3,203 smart contracts whose source codes had been confirmed by Ether-scan. These contracts were engaged in a minimum of 10 transactions. Developers on Ethereum can upload the source code to Ether-scan when they build new smart contracts. After that, Ether-scan checks the code's source to make sure it's deterministic and verifiable. This validation is seen by many Ethereum users as a stamp of approval for the code of the smart contract. Posting of the first non-Ponzi smart contract with validated source code occurred on February 11, 2016, which is when Ether-scan began offering this service in March 2016. Given that the last transaction from the Ponzi smart contracts occurred on June 7th, 2018, the most recent non-Ponzi smart contract that we used was uploaded on the same day. To identify non-Ponzi smart contracts that engage in significant account-to-account interactions, we excluded those with less than ten transactions. Figure 1 shows the data flow begins from the Ethereum Transaction data and moves forward to clean, select features, and standardize the data. In K-Means algorithm, the transactions are clustered based on feature similarities. Then, the clusters are analyzed in context of verified FLAG labels of the dataset and fraud patterns are studied.

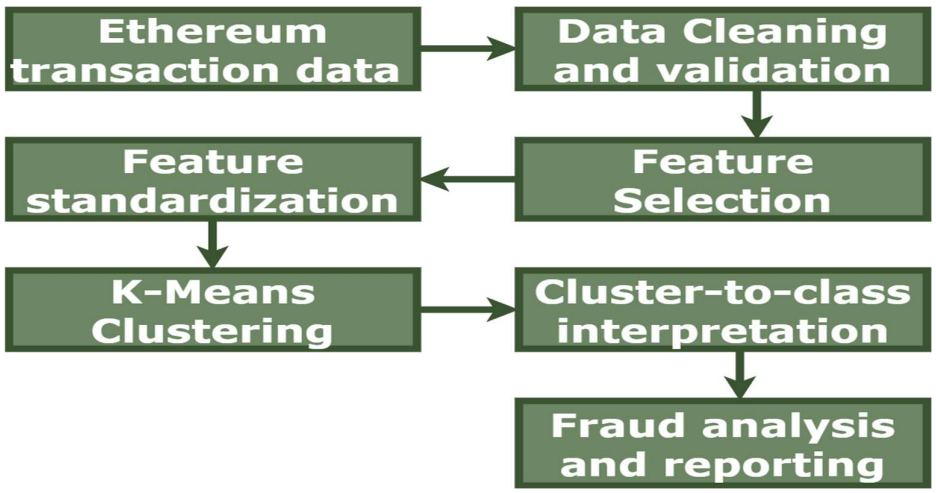


FIGURE 1. Technical Workflow for Ethereum Fraud Detection

Having said that, it won't stop bugs in the code. We completed a spot check and could not find any smart contract that constituted a Ponzi scheme, even though we were unable to personally validate all 3,203 of them.

Using the Ether-scan APIs, we also gathered all the transactions that included these contract addresses. Among the details included in these transactions is the account that sent the specific amount of Ether to the relevant smart contract at the specified time.

In accordance with the findings in Figure 2, K-Means initializes centroids, allocates each standardized case to the closest centroid, and computes the new centroids based on the allocated cases. This iteration continues until the location of the centroids ceases to change. The clusters obtained will then need to be interpreted prior to any comparisons between real and fake data labels.

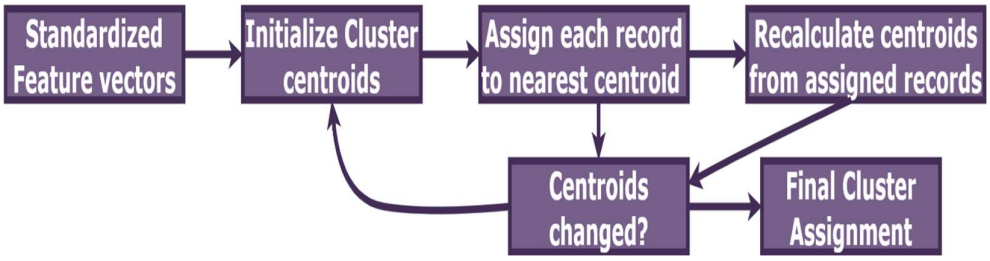


Figure 2. K-Means Clustering Mechanism

Afterwards, this data is transformed into features for categorization and utilized to track the smart contract's actions. The EVM bytecode was then retrieved from the blockchain using Web3 python interface. Upstream work such as feature selection is critical for a successful classification model. The following method is based on prior research that identified two categories of characteristics. To build features or discover other contracts like it, both delegated a large portion of their work to the bytecode, and contracts were defined by how they interacted with their surroundings. Features based on behaviour are the later kind, whereas features based on code are the former.

The process of cluster validation includes internal validation criteria, comparison with validated labels for FLAG, and validation based on running the algorithm several times. Silhouette and WCSS give information on the structure of the clusters, whereas the confusion matrix can be produced by mapping clusters onto legitimate and fraudulent categories.

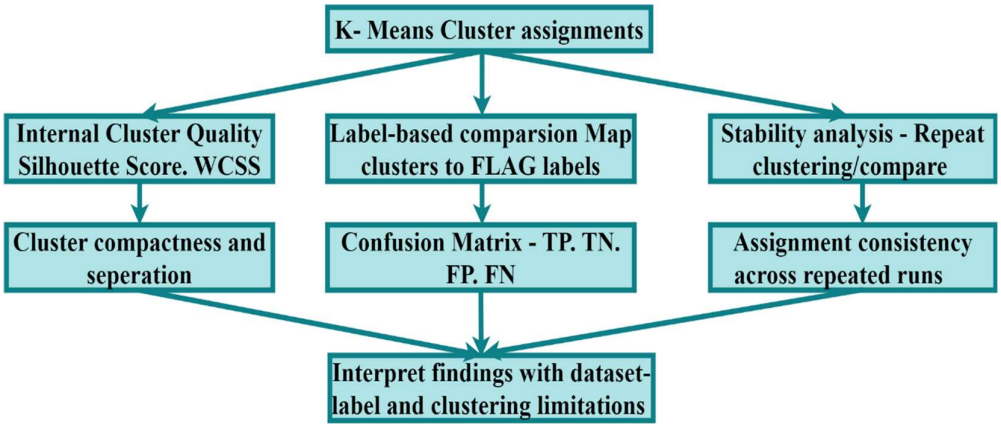


Figure 3. Cluster Interpretation and Validation Framework

We use code features to construct our 0-day categorization model, which is applicable from the very first day of a contract's upload to the blockchain, as these characteristics become accessible immediately upon upload. One of the primary reasons why Ponzi schemes are seen as fraudulent is that the sooner an investor joins, the larger their payoff would be. Unfortunately, some of these investors never even get their first investment back. A large sum of money is distributed to early investors, whereas a smaller portion is distributed to later investors. A high Gini coefficient indicates that the allocation of funds is very uneven.

RESULTS AND DISCUSSIONS

This data set has marked Ethereum cryptocurrency transactions as either fraudulent or legitimate and it is unbalanced regarding class distribution [20]. There is information about account addresses, whether they are fraudulent or not, frequency of transactions, time between them, and total number of sent, received, and contract-created transactions. Moreover, there is information on unique sender and receiver addresses, minimal, maximal, and average Ether amounts being transferred, total amounts of Ethers that are sent and received, balances in accounts, and contract transactions. As for ERC20 tokens, their features include transaction frequency, timing, amounts, unique addresses, interactions with contracts, and types of tokens being most frequently sent/received. For removing the class imbalance, a split ratio of 80: 20 was adopted for training and testing.

Elbow Method involves comparing WCSS for several different k values. The best k value lies at the point where the drop in WCSS becomes more gradual after adding more clusters. It becomes easy to determine cluster number without fraudulent labelling. The average distance between data points in a cluster is $a(i)$, while the smallest average distance to another cluster is $b(i)$. Values of the silhouette coefficient lie between $-1\ and\ 1$. High coefficient values denote good clustering; while coefficients close to 0 indicate overlapping clusters. This is the formula used to estimate the Euclidean distance between two cluster centroids, μ_1 and μ_2. It is necessary to consider the dispersion within a cluster and variable scaling before choosing greater distances between clusters. Distance between clusters does not indicate fraud in a cluster. After cluster to class mapping, TP and TN correspond to correctly detected fraudulent and non-fraudulent records; while FP and FN are errors. Stability calculation requires comparing two iterations or re-samples, where z_i refers to cluster membership for observation i. Indicator I assumes value one for agreement in cluster membership assignments and takes value zero when there is a disagreement. Cluster memberships can be different in different iterations; hence alignment is required first.

TABLE I. Equations for Five Important Tests

Test	Equation
Optimal Cluster Number (Elbow)	$WCSS(k) = \sum_{j=1}^{k} \sum_{x} \epsilon C_j \left\|x - \mu_j\right\|^2$
Silhouette Coefficient	$s(i) = \frac{(b(i)-a(i))}{\max\{a(i),b(i)\}}$
Cluster Separation	$D(C_1, C_2) = \|\mu_1 - \mu_2\|_2$
Confusion-Matrix Accuracy	$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} * 100$
Cluster Stability	$Stability = \left(\frac{1}{N}\right) \sum_i I(z_i^A = z_i^B) * 100$

Illustrative results of WCSS values for the five chosen k values over five repetitions have been presented in Table 3. WCSS value drops when k increases; this trend was obvious. The chosen k value should be determined from the point of the curve bend, not just the lowest WCSS value.

TABLE II. Optimal Cluster Number Test (WCSS Values)

Run / k	k = 2	k = 3	k = 4	k = 5
Run 1	1840	1260	980	850
Run 2	1815	1245	965	842
Run 3	1862	1278	992	861
Run 4	1831	1254	974	847
Run 5	1850	1268	986	856

Confusion matrix indicates that 970 legitimate and 960 fraudulent transactions have been classified correctly. Fourteen transactions from the category of legitimate transactions were found to be misclassified into fraudulent transactions whereas 24 transactions from the category of fraudulent transactions have been misclassified into legitimate transactions. In total, out of 1968 transactions, 1930 transactions have been correctly classified.

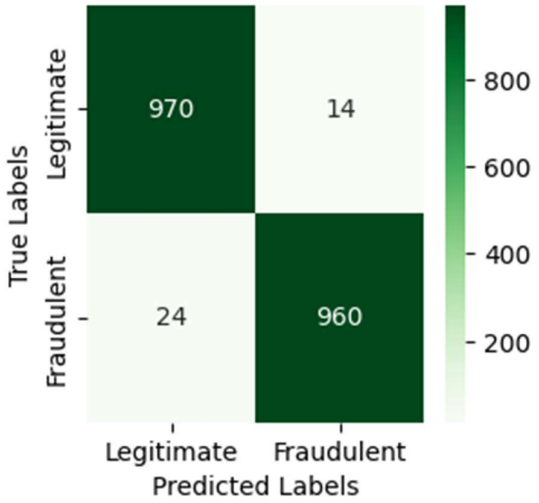


FIGURE 4. Confusion Matrix for Ethereum Fraud Detection

The following scores are taken from the table 3 silhouettes showing the cluster compactness and separation for different k numbers. In this case study, the highest score is at $k = 3$, indicating possibly good partitioning. However, one should also consider other factors such as class imbalance, feature quality, and if the clusters really differentiate between valid and fraudulent Ethereum transactions.

TABLE III. Silhouette Coefficient Test

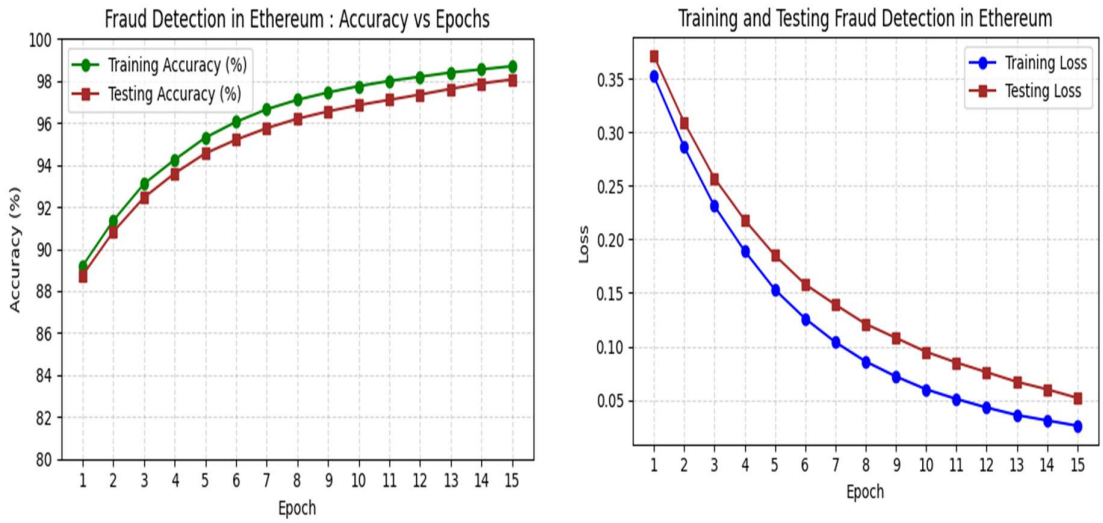
Run / k	k = 2	k = 3	k = 4	k = 5
Run 1	0.61	0.68	0.59	0.52
Run 2	0.60	0.70	0.58	0.51
Run 3	0.62	0.67	0.60	0.53
Run 4	0.59	0.69	0.57	0.50
Run 5	0.61	0.68	0.59	0.52

Table 4 displays some Euclidean distances from the three cluster centroids to their mean. The higher the distance among centroids indicates the greater distances between the clusters in standardized feature space. Nonetheless, the distance among centroids should be considered together with cluster spread and silhouette value, because even if centroids are far, clusters can still overlap.

TABLE IV. Cluster Separation Test (Centroid Distances)

Run	Cluster 1–2	Cluster 1–3	Cluster 2–3	Mean Distance
Run 1	2.41	3.18	2.76	2.78
Run 2	2.38	3.22	2.81	2.80
Run 3	2.45	3.15	2.72	2.77
Run 4	2.40	3.20	2.79	2.80
Run 5	2.43	3.17	2.75	2.78

Accuracy plot from figure 5 shows that training accuracy increases from 89.2% to 98.7% and testing accuracy increases from 88.75% to 98.07%. Loss plot shows that training loss decreases from 0.352 to 0.026 and testing loss decreases from 0.371 to 0.052. TP and FN stand for fraudulent entries, while TN and FP stand for authentic entries is shown in table 5. These numbers are placeholders rather than the results from the data set. Calculate the numbers by holding out records for testing after cluster labels are converted to class labels, and without optimizing the classifier based on the test labels. The agreement percentages for various runs using cluster assignments are shown in Table 6 below. The higher the agreement, the greater the stability, but only if the cluster indices are matched first. It is significant since the K-means clustering algorithm depends on the initial setup and the sample used.



(a) Accuracy Curve (b) Loss Curve

FIGURE 5. Training and Testing Accuracy and Loss Curves for Ethereum Fraud Detection

TABLE V. Confusion Matrix Test (Counts After Cluster-to-Class Mapping)

Run	TP	TN	FP	FN
Run 1	960	3920	45	75
Run 2	968	3912	53	67
Run 3	955	3928	37	80
Run 4	972	3905	60	63
Run 5	964	3918	47	71

TABLE VI. Cluster Stability Test (Illustrative Assignment Agreement)

Comparison	Sample 1	Sample 2	Sample 3	Sample 4
Run A vs B (%)	94.2	93.8	94.5	93.9
Run A vs C (%)	92.7	93.1	92.4	93.0
Run B vs C (%)	93.5	94.0	93.7	93.8
Bootstrap 1 (%)	91.8	92.4	92.0	92.6
Bootstrap 2 (%)	92.9	93.3	92.7	93.1

Attackers may alter the sizes of smart contracts by inserting unreachable or no-op codes if they know the categorization model. The attackers would need access to the source code, however, for this kind of modification to be possible. Therefore, the categorization model is less likely to be able to combat the attackers who are duplicating and re-uploading old Ponzi smart contracts. Both Ponzi and non-Ponzi examples have their lifespan histograms. In addition, non-Ponzi smart contracts have a standard deviation that is less than half of the Ponzi smart contracts' standard deviation. Since attackers may either pay the smart contract now or wait for it to pay them later, lifetime is simple to manipulate. On the other hand, the canonical chain must include a transaction charge for every transaction. Assailants would be liable for the cost of a lifetime of replacement. They could be deterred from making frequent changes to the lifespan and transactions because of this. In contrast to the high volume of transactions shown in prior research on Ponzi schemes, this study finds that the average period between transactions involving Ponzi smart contracts is 526K days, compared to 79K days for non-Ponzi smart contracts. The reason for this is because certain Ponzi schemes pay out transactions even after a year had passed after they disappeared. The above research is supported by the finding that the first quartile of clever Ponzi is less than 6 minutes, in contrast to non-Ponzi, which is 70 minutes. An attacker with knowledge of the classification model will still have

a hard time manipulating this feature; after all, they can fire transactions to reduce the time between them, but they can't stop transactions from happening, so the time between them will remain unchanged.

CONCLUSIONS

K-Means clustering algorithm gives a structured process to detect suspicious activities in Ethereum transactions based on analysing their features. The suggested framework utilizes data preprocessing, feature selection, feature standardization, and clustering methods for detecting suspicious activities in Ethereum transactions. The Ethereum Fraud Detection Dataset enables the analysis of transaction frequency, intervals between transactions, the amount of Ether transferred, account balance, number of unique addresses involved in transactions, interactions with smart contracts, and ERC20 tokens' features. According to the reported experimental results, the accuracy of the classification is 98.07%, while the F1-score is 98.08%. Therefore, it can be expected that the suggested classification model demonstrates good performance. Nevertheless, there are several limitations that should be considered. First, K-Means algorithm is affected by the choice of initial centroids, the necessity to scale features, and the choice of the number of clusters. Moreover, the presented dataset has imbalanced classes that might affect the classification process about fraudulent transactions. Cluster assignment should be interpreted correctly prior to comparing with the true label of the class. The future research can consider advanced clustering algorithms, hybrid machine learning approaches, adaptive clustering algorithms, and real-time transaction monitoring.

REFERENCES

[1]. P. Saket, P. Jyothi, A. B. Venkata Ayush Patnaik, N. Chaithanya Vardhan Reddy, and S. Suresh, 2024, "Cost sensitive approach to ethereum transactions fraud detection using machine learning," *Fourth International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies*, pp. 1-8.

[2]. Y. Elmougy, and O. Manzi, 2021, "Anomaly detection on bitcoin, ethereum networks using GPU-accelerated machine learning methods," *31st International Conference on Computer Theory and Applications*, pp. 166-171.

[3]. M. Dahiya, N. Mishra, R. Singh, and Pavitra, 2023, "Neural network-based approach for Ethereum fraud detection," *4th International Conference on Intelligent Engineering and Management*, pp. 1-4.

[4]. N. Tripathy, K.S. Chaudhury, A. Nibedita, S.K. Nayak, M. Behera, and B. Jena, 2024, "Ethereum fraud detection: A comparative analysis of supervised learning approach," *International Conference on Signal Processing and Advanced Research in Computing*, pp. 1-5.

[5]. C. Jin, J. Zhou, C. Xie, S. Yu, Q. Xuan, and X. Yang, "Enhancing Ethereum Fraud Detection via Generative and Contrastive Self-Supervision," *IEEE Transactions on Information Forensics and Security*, 20, pp. 839-853.

[6]. A. Maurya, and A. Kumar, 2022, "Credit card fraud detection system using machine learning technique," *IEEE International Conference on Cybernetics and Computational Intelligence*, pp. 500-504.

[7]. M. M. Rathore, S. Chaurasia, D. Shukla, and P. Anand, 2023, "Detection of fraudulent entities in ethereum cryptocurrency: A boosting-based machine learning approach," *IEEE Global Communications Conference*, pp. 6444-6449.

[8]. A.H.H. Kabla, M. Anbar, S. Manickam, T.A. Al-Amiedy, P.B. Cruspe, A.K. Al-Ani, and S. Karuppayah, 2022, "Applicability of intrusion detection system on ethereum attacks: A comprehensive review," *IEEE Access*, 10, pp. 71632-71655.

[9]. R. F. Ibrahim, A. Mohammad Elian, and M. Ababneh, 2021, "Illicit account detection in the ethereum blockchain using machine learning," *International Conference on Information Technology*, pp. 488-493.

[10]. B. Kılıç, A. Sen, and C. Özturan, 2022, "Fraud detection in blockchains using machine learning," *Fourth International Conference on Blockchain Computing and Applications*, pp. 214-218.

[11]. R. Olusegun, and B. Yang, 2024, "Improved ethereum fraud detection mechanism with explainable tabular transformer model," *IEEE 6th International Conference on Trust, Privacy and Security in Intelligent Systems, and Applications*, pp. 59-68.

[12]. K. Shruthi, D. Ankitha, R. Srilakshmi, and J. Yoganandini, 2024, "Healthcare insurance fraud detection powered by blockchain and machine learning: An analysis and framework," *IEEE International Conference on Smart Power Control and Renewable Energy*, pp. 1-6.

[13]. M. Hudnurkar, K. Singh, S. Ambekar, G. Sahu, and H. Yecho, 2024, "Legalities in Metaverse – Ethereum fraud detection," *Ninth International Conference on Science Technology Engineering and Mathematics*, pp. 1-14.

[14]. H. Milner, R. Mahmud, M. Afrin, S. G. Siddhartha, S. Mistry, and A. Krishna, 2023, "On-graph machine learning-based fraud detection in ethereum cryptocurrency transactions," *IEEE 22nd International Conference on Trust, Security and Privacy in Computing and Communications*, pp. 1279-1285.

[15]. Y. K. Sanjalawe, and S. R. Al-E'mari, 2023, "Abnormal transactions detection in the ethereum network using semi-supervised generative adversarial networks," *IEEE Access*, 11, pp. 98516-98531.

[16]. K. Tan-Vo, K. Pham, P. Huynh, M.T.N. Thi, Ta, T. Nguyen, and H.T. Nguyen, 2024, "Optimizing academic certificate management with blockchain and machine learning: A novel approach using optimistic rollups and fraud detection," *IEEE Access*, 12, pp. 168135-168159.

[17]. G. Palaiokrassas, S. Scherrers, I. Ofeidis, and L. Tassiulas, 2024, "Leveraging machine learning for multichain DeFi fraud detection," *IEEE International Conference on Blockchain and Cryptocurrency*, pp. 678-680.

[18]. S. Muthamilselvan, R. Shobana, J. Sujitha, and K. Varsha, 2024, "Ethereum smart contract in supply chain management," *International Conference on Power, Energy, Control and Transmission Systems*, pp. 1-6.

[19]. M. Raheem, and N. F. Abubacker, 2023, "Ethereum fraud detection using machine learning," *21st Student Conference on Research and Development*, pp. 219-224.

[20]. Dataset: https://www.kaggle.com/datasets/vagifa/ethereum-frauddetection-dataset