INTERNATIONAL JOURNAL OF SOFTWARE ENGINEERING AND COMPUTER SYSTEMS
E-ISSN: 2180-0650
VOLUME 12, ISSUE 1, 2026, pp. 50-59
DOI: https://doi.org/10.15282/ijsecs.12.1.2026.5.0151

RESEARCH ARTICLE

Machine Learning Approach to Ethereum Fraud Detection: Comparative Analysis of Logistic Regression and Random Forest

Nathania Vannesa[1]*, Nusrat Zahan Nisha[2], Wara Shindi Shella May Wara[1]

[1]Data Science, Faculty of Computing, Universitas Pembangunan Nasional "Veteran" Jawa Timur, 60294 Surabaya, Indonesia
[2] Faculty of Computing, Universiti Malaysia Pahang Al-Sultan Abdullah, 26600 Pahang, Malaysia

ABSTRACT - Ethereum's pseudonymous and decentralized nature has made it challenging to trace fraud since its inception. The fraud type identified is fraudulent reuse of the same funds from copied unique received addresses and a different time between the first and last user transaction. This paper is concerned with comparing and examining the performance of Logistic Regression and Random Forest models in detecting unusual Ethereum transactions and finding the most significant features influencing these outcomes. Traditional approaches have a tendency to overlook lightweight, interpretable models that may be utilized for real-time filtering, or they focus on Bitcoin rather than Ethereum. In order to fill this gap, this study utilizes machine learning to detect meaningful patterns of behaviour related to fraud. This study compares the performance of Random Forest and Logistic Regression models in classifying fraudulent Ethereum transactions. The dataset is extracted from a publicly available repository called Ethereum Fraud Detection. The dataset was normalized and cleaned using MinMaxScaler, and then split into 80% training and 20% testing subsets. Feature scaling, correlation analysis, and removal of duplicate variables (such as ERC20 tokens) were part of the preprocessing. With 90.40% accuracy and F1-score of 89.17%, the model suggests that Random Forest is a better performer than Logistic Regression. The ability of the model to identify non-fraud instances is clear from the visualization of the confusion matrix, which also identifies areas of improvement in identifying actual fraud cases. These results give business organizations actionable recommendations on how to deploy real-time detection, rank high-risk transaction signals, and utilize adaptive machine learning architectures. The paper promotes further research into anomaly detection based on deep learning and demonstrates the potential of feature-based machine learning in bolstering Ethereum security infrastructure.

ARTICLE HISTORY
Received : 27 May 2025
Revised : 19 May 2026
Accepted : 04 June 2026
Published : 09 June 2026

KEYWORDS
Ethereum Fraud Detection
Blockchain Security
Machine Learning
Random Forest
Logistic Regression

1.0 INTRODUCTION

The rapid growth of blockchain technology, particularly Ethereum, has revolutionized digital transactions. However, the innovative frameworks also present unique challenges, particularly in the realm of fraud detection. According to the official Ethereum website, Ethereum is a decentralized blockchain platform that enables developers to build and deploy smart contracts and decentralized applications [1]. Major cryptocurrencies like Bitcoin and Ethereum are experiencing severe devaluation in 2022; their prices have dropped sharply in a short period of time [2]. While Bitcoin primarily facilitates currency exchange, Ethereum was designed as a programmable platform for smart contracts and decentralized applications. Ethereum introduces a programmable blockchain framework that supports the development of smart contracts and decentralized applications. Blockchain is a distributed database that records and shares transaction data across multiple network participants [2]. Ethereum records transactions on a distributed ledger where validated transactions are immutable and cannot be altered or deleted by a single party. This immutability enhances the integrity and security of blockchain records by preventing unauthorized modifications. Consequently, attackers cannot easily manipulate historical transaction data without gaining control of a substantial portion of the network's computational resources. These characteristics make Ethereum a secure platform for decentralized applications and digital transactions. This flexibility has made Ethereum a cornerstone of blockchain innovation, enabling developers to create versatile and efficient solutions for various industries.

Fraud detection on Ethereum poses unique challenges due to its decentralized and pseudonymous nature. In May 2024, Ars Technica reported that MIT students exploited a flaw in Ethereum's MEV-boost software to steal $25 million within seconds. According to the prosecution, they accomplished this by taking advantage of a flaw in the MEV-boost software code, which is utilized by the majority of Ethereum network "validators" [3, 4]. Traditional fraud detection methods, often designed for centralized systems, fall short when applied to blockchain networks. In Ethereum, transactions occur on a global, distributed ledger, making it difficult to rely on conventional oversight mechanisms. This gap is further complicated by evolving regulatory landscapes, such as the Market in Crypto-Assets (MiCA) regulation,

***CORRESPONDING AUTHOR | Nathania Vannesa** | ✉ vannesanath@gmail.com

effective July 2024, which aims to standardize protection, and market stability [5]. The open nature of the platform allows malicious actors to exploit vulnerabilities and conduct fraudulent activities, such as unauthorized transactions or manipulation, without immediate detection. Identifying fraudulent transactions in this environment requires advanced analytical techniques capable of processing large volumes of data and uncovering hidden patterns indicative of fraud.

The underlying problem addressed in this study is the insufficiency of traditional fraud detection techniques within the framework of decentralized, high dimensional blockchain networks like Ethereum. Traditional approaches often lack scalability, interpretability, and the ability to act upon fraud attempts in a real time manner. Most existing work either focused on specific fraud types in artificial simulations rather than actual transactional data. In decentralized systems, the absence of a central authority makes manual inspection or pre-defined rules-based fraud detection infeasible. Therefore, automatic intelligent systems capable of examining huge volumes of blockchain data, extracting subtle patterns, and making accurate classification of fraudulent behaviour are in pressing demand.

This paper makes several novel contributions to the existing literature on blockchain fraud detection. The current works tend to deal with smart contract-based fraud or general cryptocurrency networks; this research explicitly deals with Ethereum transactional data. Second, while current studies point to complex models like deep learning which tend to sacrifice interpretability for performance, this research uses interpretable machine learning models that can directly explain how decisions are made. Third, the utilization of the PySpark for training and testing models ensures that the method is not only scientifically but also realistically feasible for implementation in real-time fraud detection systems and contributing to a more secure Ethereum ecosystem while identifying key features associated with fraudulent transactions. Furthermore, this study provides an interpretable comparison between Logistic Regression and Random Forest models using real Ethereum transaction data, enabling practical deployment for fraud monitoring systems.

The rest of the paper is organized as follows: Section 2 is a review of relevant research works on blockchain fraud detection with their strength, limitations, and relevance to this study. Section 3 describes the methodology, including data preprocessing, model implementation, and performance metrics. Section 4 gives the result and comparative performance of the models. Finally, Section 5 concludes the paper and outlines future research directions.

2.0 RELATED WORKS

As reported by Elmougy and Manzi, in the contexts of implementation of Deep Learning on Ethereum Detection [6]. This work investigates classification algorithms to obtain the best-performing model with high precision and high recall. The classification algorithms used in the study include J48, Random Forest, and Stochastic Gradient Descent. The authors focus on the precision and recall of Ponzi instances. The feature selection to have an efficient classification model that authors choose are SSTORE, POP, MSTORE, SWAP1 [7]. The statistical comparison showed that Random Forest achieved the highest recall and F-score among the evaluated models. However, SGD achieved a precision of 0.98 and an F-score of 0.96. SGD resulted in better precision than the others. The authors demonstrated that the proposed model can effectively identify Ponzi schemes by flagging suspicious smart contracts during deployment on the blockchain. The precision of 0.99 and recall 0.97 gain from full-feature.

Koa et al. [8] proposed an Ethereum-based decentralized Public Key Infrastructure (PKI) framework that improves trust management through a reward-and-punishment mechanism based on a Web of Trust model. Using custom ERC-20 tokens (PKIToken) for stable incentives, the system requires multiple trust/untrust signatures (adjustable thresholds) to validate identities, reducing Sybil attacks. Simulations with "Good," "Bad," and "Normal" nodes demonstrate its effectiveness: bad actors lose tokens, while honest participants maintain balances, outperforming ETHERST 2.0. This work highlights blockchain's potential to decentralize trust in PKI systems, though real-world scalability remains unexplored. The research focuses on simulation-based evaluation with different type of node. The study contributes to the advancement of innovative blockchain-based PKI solutions.

The work in [9] proposed a machine learning framework for detecting fraudulent transactions in Bitcoin and Ethereum networks. Using GPU-accelerated algorithms such as Support Vector Machine (SVM), Random Forest, and Logistic Regression, the study analysed large-scale blockchain transaction metadata and achieved strong fraud detection performance. The models achieved strong performance, with 96.9% accuracy and 0.987 recall for Bitcoin, and 80.2% accuracy with 0.835 recall for Ethereum. Through sensitivity analysis, the study provides insights into key fraud indicators while demonstrating the potential for real-time anomaly detection. The methodology proves adaptable beyond cryptocurrencies, offering applications for healthcare, government, and financial blockchain systems seeking to enhance security against malicious actors. This work advances blockchain integrity by combining technical innovation with empirical validation at scale.

This paper addresses transactional fraud in Ethereum using interpretable machine learning models, namely Random Forest and Logistic Regression. This work uniquely focuses on Ethereum transactions and provides a comparative performance evaluation using a real-world dataset. The models are implemented in a PySpark environment that supports scalable real-time data processing. In addition to standard evaluation metrics, this paper also focused on the top predictive features that influence fraud detection, delivering both utility and interpretability.

The field of focus, strength, and limitation will be explained on the Table 1 below;

Table 1. Related Works

Study	Focus area	Strength	Limitations	How this study improves?
[6] Elmougly and Manzi	Ponzi scheme identification using smart contract.	High precision and recall, comparison of deep learning model.	Limited to smart contract information, not real transaction level.	Focused on real Ethereum transaction, not only Ponzi.
[8] Koa et al.	Authentication of identities.	New WoT mechanism, ERC20 token-based simulation	Simulation only, not tested on actual Ethereum Data	Uses actual transaction data and interpretable Machine Learning.
[9] Eunjin Jung	General fraud detection using large scale machine learning	Massive dataset, GPU acceleration, multi algorithm	Does not encompass Ethereum-specific findings, model interpretability is restricted.	Ethereum specific focus, explain feature importance
Proposed Method	Ethereum transactional-level fraud detection.	Real dataset, interpretable models, scalable with PySpark	Baseline comparison only.	Balances accuracy and interpretability for real-world deployment

Compared with previous studies that focus on Ponzi schemes, smart contracts, or general cryptocurrency fraud, this study focuses specifically on Ethereum transaction-level fraud detection using interpretable and scalable machine learning models.

3.0 METHODS AND MATERIAL

This project focuses on leveraging machine learning techniques to address the issue of fraud detection within the Ethereum blockchain. The scope includes the following key aspects: Data Analysis and Preparation, Exploratory Data Analysis, Feature Importance Analysis, and Visualization.

3.1 Meta Data

The dataset is sourced from a publicly available repository titled 'Ethereum Fraud Detection' [21]. The data consists of 9841 rows and 51 features, the data type of 39 columns is decimal, 9 columns are integer, and 3 columns are string. Metadata, defined as structured information that describes, explains, or simplifies the retrieval and management of data [10], plays a critical role in ensuring the reproducibility and scalability of machine learning workflows, particularly in blockchain analytics [11]. The dataset consists of 9841 Ethereum transactions, out of which roughly 1500 are labeled as fraudulent and 7000 as non-fraudulent. Since the dataset is imbalanced, future work should apply class-balancing techniques such as SMOTE, class weighting, or cost-sensitive learning to improve fraud detection performance. The focus of this research is to predict transactions that indicate fraud, so the features used are numeric types. Prior studies emphasize that metadata in blockchain contexts often requires preprocessing to handle high dimensionality and heterogeneity [12].

The variables used will be explained on Table 2:

Table 2. Features

No	Variable	Description
1.	FLAG	Whether or not there is fraud in the transaction.
2.	Avg min between sent tnx	Average number of minutes between account transactions sent.
3.	Avg min between received tnx	Average time in minutes between received transactions.
4.	Time Diff between first and last (Mins)	The time difference between the first and last transaction.
5.	Sent tnx	The total quantity of regular transaction sent.
6.	Received Tnx	The total quantity of regular transactions received.
7.	Number of Created Contracts	The total number of contract transactions that have been created.
8.	Unique Received from Addresses	Total unique addresses from which transactions were received by the account.
9.	Unique Sent to Address	The total number of unique addresses that transactions were sent from an account.
10.	Min Value Received	Minimum value in Ether received.
11.	Max Value Received	The highest amount of Ether ever received.
12.	Avg Val Received	The average amount of Ether that was ever received.
13.	Min Val Sent	The lowest amount of Ether ever transmitted.
14.	Max Val Sent	The highest amount of ether ever transmitted.

15.	Avg Val Sent	The mean amount of ether sent.
16.	Min Value Sent to Contract	The minimum amount of ether sent to a contract.
17.	Max Val Sent to Contract	The highest amount of Ether that can be transmitted to a contract.
18.	Avg Value Sent to Contract	Average Ether value sent to contract.

3.2 Data Analysis and Preparation

The implementation of the model and feature scaling in this research will be carried out using PySpark which will make it possible to perform real-time processing of large-scale data in a distributed environment using Python [13].

The algorithm presented in Table 3 shows the steps required to train and evaluate the model.

Table 3. Algorithm for Ethereum Fraud Detection

No	Step
1.	Import the required libraries on Pyspark; SparkSession, MinMaxScaler, seaborn, etc.
2.	Load the data into a variable
3.	Clean the data from missing values and duplicate values.
4.	Drop irrelevant features
5.	Implementation of normalization using MinMaxScaler
6.	Split the dataset into 80% training data and 20% of testing data
7.	Correlation between features
8.	Implementation of Logistic Regression Model
9.	Evaluate Logistic Regression using MulticlassClassificationEvaluator
10.	Implementation of Random Forest
11.	Evaluate Random Forest using MulticlassClassificationEvaluator.
12.	Confusion Matrix for both model
13.	Visualization top 10 important features

Non-predictive identifiers and irrelevant ERC20-related fields were removed to reduce noise and prevent overfitting. ERC20 is not valuable for this implementation of the machine learning model because ERC20 only focuses on a token standard on the Ethereum blockchain.

In this experiment, ERC20-related features, as well as the variables ‘Unnamed:0’, ‘Index’, ‘_c0’, and ‘Address’, were removed from the dataset because they are non-predictive identifiers and do not contribute to fraud classification. Missing values were addressed by either imputation with the mean or removal. In this observation, *dropna* was implemented to remove the missing value. After cleaning the data from missing value, dropping the duplicate data also important. The cleaned data perform as in table 4:

Table 4. Amount of Clean Data

Item	Status	Amount of Data	
		Before	After
Missing Value	No	9841	9841
Duplicate Value	Yes	9841	9268

Normalization is the process to organize the dataset to reduce redundancy and improve the data integrity. We can normalize MinMaxScaler using Pyspark by importing *pyspark.ml.feature import MinMaxScaler.*

The result performs as this table 5:

Table 5. Scaled Features of Sent and Received Tnx

Avg Min Between Sent Tnx (Original)	Avg Min Between Received Tnx (Original)	Avg Min Between Sent Tnx (Scaled)	Avg Min Between Received Tnx (Scaled)
1.24	0.07	0.000	0.0000
0.00	2,338.02	0.000	0.0048
0.00	1,689.37	0.000	0.0035
0.00	0.00	0.000	0.0000
32,673.70	54,225.86	0.0759	0.1125

The scaling process successfully normalizes the raw transaction time metrics from the original scattered values into a uniform range between 0 and 1. This standardized presentation in table 5 ensures that both features are systematically prepared for unbiased model training.

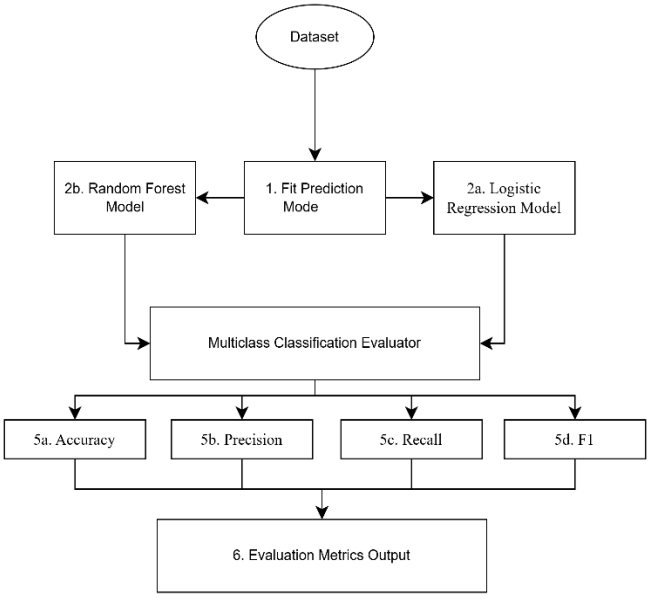

Figure 1. Flowchart

The algorithm represented in Figure 1 shows the steps required to fit the prediction model from the dataset based on logistic regression model and random forest model. This research is going to use an evaluator referred to as Multiclass Classification Evaluator. Accuracy, precision, recall, and F1 are four evaluators that are used. Each one is for a specific purpose. Accuracy indicates that the model was properly classified. Recall indicates how good the model is in capturing most of the actual data. F1 measures the balanced ratio between precision and recall.

3.3 Modeling

Comprehensive EDA is conducted to identify distribution patterns and feature correlations, providing valuable insights into the data structure. Exploratory Data Analysis is critical for understanding underlying data characteristics and guiding model selection, particularly in high-dimensional financial datasets [14]. The project applies both logistic regression and random forest models to classify transactions. These models are chosen for their ability to handle structured data and provide interpretable results. Exploratory Data Analysis will cover correlation analysis, logistic regression, and random forest to predict the fraud transaction on Ethereum.

Logistic Regression is applied as a baseline model to predict a binary result of fraudulent transactions. While simple, it provided initial insights into the data. The model produces a binary output representing fraudulent (1) or non-fraudulent (0) transactions. A Logistic function is used in logistic regression to model the connection between the predictor variables and the binary result. Logistic Regression is a model well-suited to data mining challenges, including multicollinearity, missing values, and redundant features [15]. The logistic function commonly derived from mathematical calculation of the formula in (1) below;

$$E[\, y_i = 1 \,|x_i, \beta] = pi = \frac{e^{x_i \beta}}{1 + e^{x_i \beta}} = \frac{1}{1 + e^{-x_i \beta}} \, , for \; i = 1, \ldots \ldots n \tag{1}$$

The equation represents the probability function of logistic regression, y_i is the dependent variable or response variable. It represents the binary outcome for observation i and x_i is the independent variable or feature vector that represents from input feature for observation *i.* For the last symbol is β that represents as a coefficient vector (parameter vector) that associated with each feature in x_i. This model is robust to collinearity and missing data [16], making it suitable for preliminary fraud detection [17]. However, its linearity assumption limits performance on complex non-linear patterns inherent in blockchain transaction.

Random Forest, an ensemble learning method, is suitable for both classification and regression tasks. A more robust and flexible model, random forest provided higher accuracy and better feature importance insights. Random Forest can handle large amounts of data efficiently and can handle imbalance datasets better rather than Logistic Regression. Imbalanced data is one of those subtle but important problems that are easy to ignore until the model begins to prefer one class over another [18]. Random Forest has proven particularly effective in detecting subtle patterns in transaction timing [19]. The value of a random vector sampled independently and with the same distribution for every tree in the forest determine the value of each tree in a random forest, which is a combination of tree predictors [20]. For the Random Forest classifier, the model was defined to have a total of 20 trees and two classes of outputs.

4.0 RESULTS AND DISCUSSION

Correlation analysis is an analysis to perform some relationships between one feature to another. The correlation value is in the range between -1 until 1. If the features have a really good relationship so it will show as 1 or in the Correlation Matrix it will be shown as Red. The result from the code is the headmap relation between variables on the data.

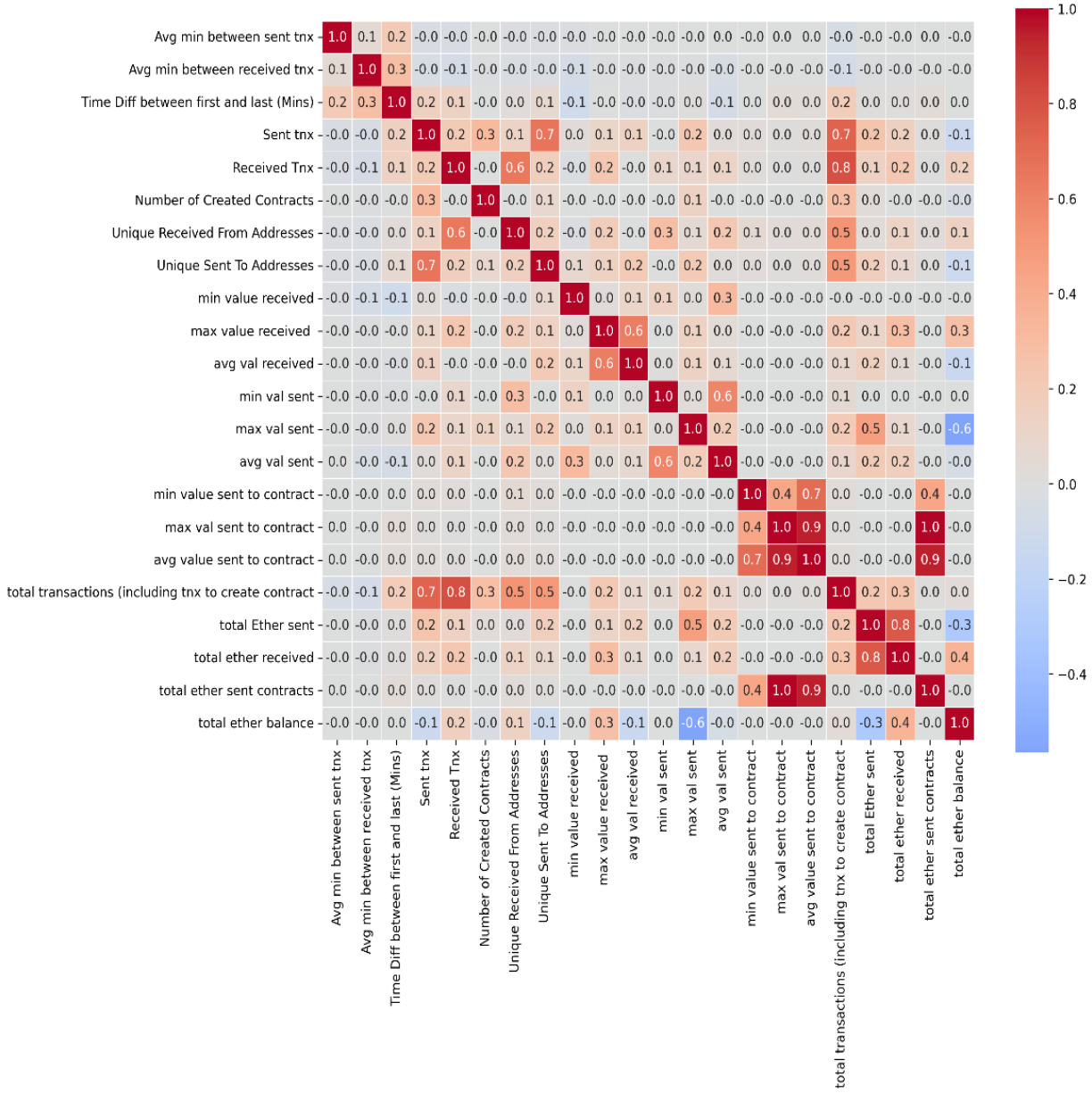

Figure 2. Correlation Matrix of Features

As the correlation matrix above, Sent transaction (Sent tnx) and Total transaction (including tnx to create contract) reach 0.7 out of 1 and Received Transaction and Total transaction reaching 0.8 out of 1. Both sent transaction and received transaction have a strong relationship on total transaction (including tnx to create contract). So, it indicates that the value of total transaction is highly dependent on received transaction and sent transaction.

The result of accuracy, precision, recall and F1 based on logistic regression model perform as in table 6:

Table 6. Performance of Logistic Regression

Metrics	Result
Accuracy	81.04%
Precision	66.42%
Recall	81.04%
F1	73.00%

Based on the result of the evaluation Logistic Regression achieved an accuracy of 81.04% indicating that the model correctly classified the majority of transactions. The precision of 66.42% indicates the proportion of transactions predicted as fraud that were correctly classified. The recall, also at 81%, demonstrates the model's effectiveness in capturing most of the actual data. The F1 score of 73% reflects a balanced trade-off between precision and recall. To visualize the distribution of fraud predictions, a confusion matrix was used.

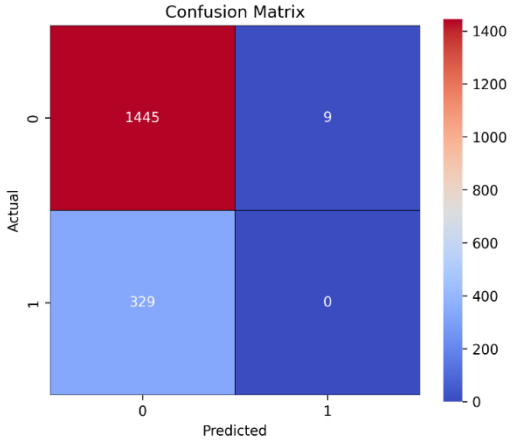

Figure 3. Confusion Matrix of Logistic Regression

The confusion matrix for Logistic Regression summarizes the predicted classifications against the actual classes. Logistic Regression demonstrated strong performance in identifying non-fraudulent transactions, producing 1,445 true negatives and only 9 false positives. However, Logistic Regression performed poorly in identifying fraudulent transactions, producing 329 false negatives and no true positive detections.

The result of accuracy, precision, recall and F1 based on random forest model perform as in table 7:

Table 7. Performance of Random Forest

Metrics	Result
Accuracy	90.40%
Precision	90.89%
Recall	90.40%
F1	89.17%

Based on the result of the evaluation Random Forest achieved an accuracy of 90.40% indicating that the model correctly classified the majority of transactions in the dataset. The precision of 90.89% indicates that most transactions classified as fraudulent were correctly identified. The recall of 90.4% demonstrates the model's effectiveness in identifying a large proportion of actual fraud cases. The F1 score of 89.17% reflects a balanced trade-off between precision and recall. Compared to Logistic Regression, Random Forest achieved the best performance for Ethereum fraud detection among the evaluated models. The model's ability to generalize was also verified by testing accuracy on both training and test sets. Training accuracy was 91%, which indicates that the Random Forest model is not overfitted and maintains high prediction consistency on unseen data. To visualize the distribution of fraud predictions, a confusion matrix was employed.

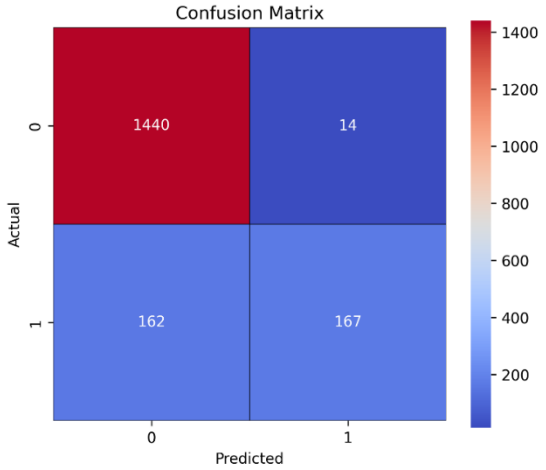

Figure 4. Confusion Matrix of Random Forest

The confusion matrix for Random Forest summarizes the predicted classifications against the actual classes. Random Forest demonstrated strong predictive performance for both fraudulent and non-fraudulent transactions The model produced 1,440 true negatives and 167 true positives. However, it also recorded 14 false positive and 162 false negatives. Although Random Forest achieved high overall accuracy, the number of false negatives indicates that further optimization is needed to improve fraud detection sensitivity.

The project emphasizes identifying key features that significantly influence fraud detection, offering practical insights into the behaviour of fraudulent transactions.

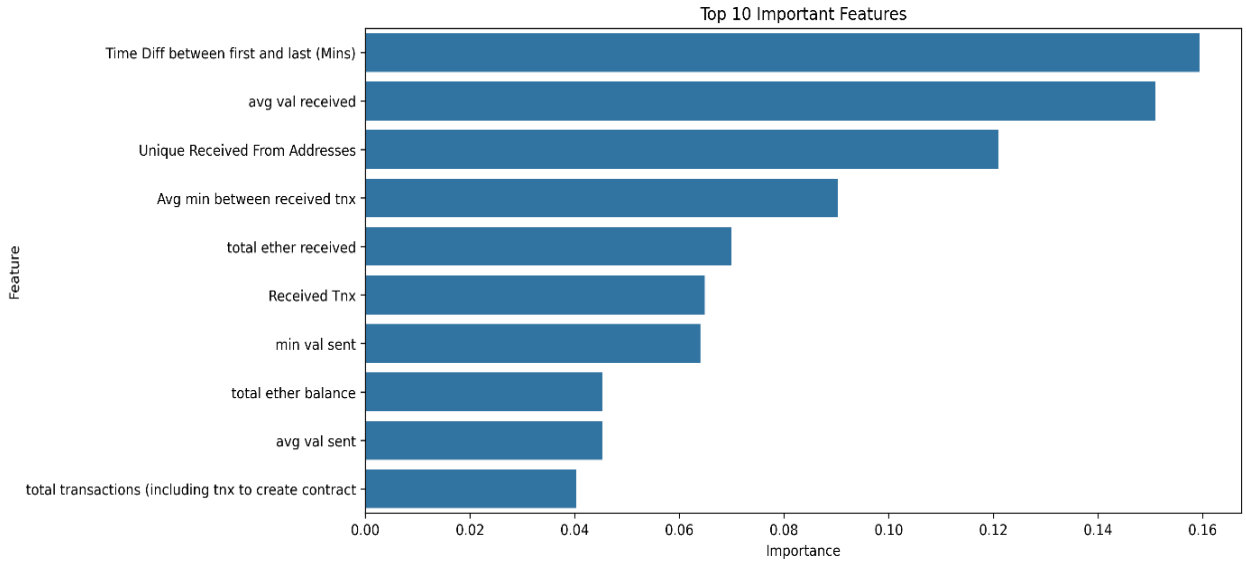

Figure 5. Top 10 Important Features

Figure 5 presents the top ten features influencing suspicious Ethereum transactions, such as ‘Unique Received Address’, ‘Time Diff between first and last (Mins)’, ‘average val received’, ‘total Ether sent’, ‘total ether received’, ‘average minimum between received transaction’, ‘minimum val sent’, ‘total transaction’, ‘received transaction’, ‘minimum value received’ which can guide fraud detection by prioritizing these metrics for monitoring.

Strong behavioural and transactional patterns connected to fraudulent operations are shown by the top 10 attributes that have been identified as crucial for Ethereum fraud detection, including the Unique Received from Address, Time Diff between First and Last Transactions, Total Ether Sent/Received, and Average/Min Values. According to these findings, companies ought to give top priority to real-time monitoring systems in order to identify irregularities like abrupt increases in Ether volume or quick transactions involving several distinct addresses. Risk scoring can be automated by implementing the Random Forest model, which has a 90% accuracy rate. Additionally, establishing thresholds for transaction quantities and periodicity helps reduce risks such as dusting attacks and money laundering. Businesses should combine these technological solutions with user education (such as confirming addresses) and cooperative industry initiatives to standardize fraud protection in order to strike a balance between security and usability. Businesses may develop a proactive, flexible approach that improves detection rates, lowers false positives, and increases confidence in Ethereum-based transactions by concentrating on three crucial characteristics. In order to combat new fraud strategies, future developments might investigate deep learning for subtle pattern recognition and dynamic model retraining.

Table 8. Evaluator of Random Forest

Feature	Risk Indicator	Mitigation Strategy
Unique Received Address	A high number of unique received addresses may indicate potential money laundering behaviour.	Limit transactions from new addresses temporarily
Time Different (First/ Last)	Short time gaps may indicate automated fraudulent activity	Require MFA for rapid sequential transactions
Total Ether Sent/ Received	Sudden spikes are theft/scams	Freeze accounts pending manual review
Min/ Max Values	Micro-transactions are probing attacks	Block low-value repetitive transactions

The best model of Ethereum Fraud detection is Random Forest with 90% accuracy and 89% of F1. Unique Received From Address, Time Diff between first and last (Mins), average val received, total Ether sent, total Ether received, average minimum between received transaction, minimum val sent, total transaction, received transaction, minimum value received are the features that are significant predictors of fraud. Suspicious financial transactions are heavily influenced by specific patterns and behaviour, such as the number of unique addresses involved. This observation highlights the critical need for companies using Ethereum for their transactions to implement robust monitoring systems to detect and prevent fraud effectively.

5.0 CONCLUSIONS

This study successfully demonstrated the potential of machine learning in detecting fraudulent transactions on the Ethereum blockchain. Logistic Regression and Random Forest were employed as the primary methods for classification, and by leveraging these models, significant progress was made in identifying patterns associated with fraud. With 90.4% accuracy and F1-score of 89.17%, the model suggests that Random Forest is a better performer than Logistic Regression. Although Random Forest outperformed Logistic Regression, the model still produced false negatives in fraud detection. Future work should therefore explore class balancing, anomaly detection, deep learning, and real-time adaptive model retraining. In addition, ensemble and deep learning approaches may be investigated to further improve fraud detection sensitivity while maintaining acceptable interpretability. The insights and models developed through this project provide a foundation for more secure and trustworthy Ethereum transactions. The insights from Table 8 indicate that specific behavioural patterns, such as the number of unique addresses involved or time gaps between transactions, play a significant role in identifying suspicious activities. For businesses, these findings emphasize the importance of focusing fraud detection efforts on high-risk categories and leveraging key transactional features to flag anomalies in real-time. By doing so, companies can not only safeguard their platforms against fraudulent activities but also ensure a secure and trustworthy environment for their customers. This proactive approach enhances customer confidence, making them feel safer and more comfortable conducting transactions, which ultimately fosters customer loyalty and supports business growth.

ACKNOWLEDGEMENTS

This study was not supported by any grants from funding bodies in the public, private, or not-for-profit sectors.

AUTHORS CONTRIBUTION

Nathania Vannesa (Conceptualization; Data curation, Writing – original draft, Coding)

Nusrat Zahan Nisha (Writing – review & editing; Project administration)

Wara Shindi Shella May (Validation; Supervision)

CONFLICT OF INTEREST

The authors declare no conflicts of interest.

REFERENCES

[1] Ethereum.org. What is Ethereum? [Internet]. 2025 [cited 2025 Feb 28]. Available from: https://ethereum.org/en/what-is-ethereum/

[2] Huang Q. Ethereum: Introduction, expectation, and implementation. Highlights Sci. Eng. Technol., 2023;39:1–7. doi: 10.54097/hset.v39i.6188.

[3] Belanger A. MIT students stole $25M in seconds by exploiting ETH blockchain bug, DOJ says. Ars Technica [Internet]. May 16, 2024 [cited 2025 Feb 28]. Available from: https://arstechnica.com/tech-policy/2024/05/sophisticated-25m-ethereum-heist-took-about-12-seconds-doj-says/

[4] Isidore C. Two former MIT students charged with stealing $25 million of crypto in 12 seconds, CNN Business [Internet]. May 16, 2024 [cited 2025 Feb 28]. Available from: https://edition.cnn.com/2024/05/16/investing/mit-crypto-hack/index.html

[5] European Parliament. Regulation (EU) 2023/1114 of the European Parliament and of the Council on markets in crypto-assets (MiCA). Off. J. Eur. Union [Internet]. 2023 [cited 2025 Feb 28]. Available from: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32023R1114

[6] Elmougy Y, Manzi O. Anomaly detection on Bitcoin and Ethereum networks using GPU-accelerated machine learning methods. In Proc. IEEE Int. Conf. Comput. Theory Appl. (ICCTA), 2021, doi: 10.1109/ICCTA54562.2021.9916589.

[7] Onu IJ, Omolara AE, Alawida M, Abiodun OI, Alabdultif A. Detection of Ponzi scheme on Ethereum using machine learning algorithms. Sci. Rep., 2023;13:18403, doi: 10.1038/s41598-023-45275-0.

[8] Koa CG, Heng SH, Chin JJ. New Ethereum-based distributed PKI with a reward-and-punishment mechanism. Blockchain: Research and Applications. 2025;6(1):100239. doi: 10.1016/j.bcra.2024.100239.

[9] Siddamsetti S, Srivenkatesh M. Efficient fraud detection in Ethereum blockchain through machine learning and deep learning approaches. Int. J. Recent Innov. Trends Comput. Commun., 2023;11(11):71–82, doi: 10.17762/ijritcc.v11i11s.8072.

[10] Zeng ML, Qin J. Metadata, 3rd ed. Chicago, IL, USA: Neal-Schuman/ALA Editions, 2022.

[11] Ashfaq T, Khalid R, Yahaya AS, Aslam S, Azar AT, Alsafari S, Hameed IA. A machine learning and blockchain based efficient fraud detection mechanism. Sensors, 2022;22(19):7162, doi: 10.3390/s22197162.

[12] Azad P, Akcora CG. Machine learning for blockchain data analysis: Progress and opportunities. IEEE Access, 2021;9:76900–76917, doi: 10.1109/ACCESS.2021.3082325.

[13] Apache Software Foundation. PySpark Overview [Internet], 2025 [cited 2025 Feb 28]. Available from: https://spark.apache.org/docs/latest/api/python/index.html

[14] Isa IGT, Zulkarnaini Z, Novianti L, Elfaladonna F, Agustri S. Exploratory data analysis (EDA) dalam dataset penerimaan mahasiswa baru Universitas XYZ Palembang. Smart Comp: Jurnalnya Orang Pintar Komputer, 2023;12(3):600–609, doi: 10.30591/smartcomp.v12i3.4125.

[15] Jiang S, Josse J, Lavielle M. Logistic regression with missing covariates—Parameter estimation, model selection and prediction within a joint-modeling framework. Comput. Stat. Data Anal., 2020;145:106907, doi: 10.1016/j.csda.2019.106907.

[16] Gu Z, Dib O. Enhancing fraud detection in the Ethereum blockchain using ensemble learning. PeerJ Comput. Sci., 2025;11:2716, doi: 10.7717/peerj-cs.2716.

[17] Aziz RM, Baluch MF, Patel S, Ganie AH. LGBM: A machine learning approach for Ethereum fraud detection. Int. J. Inf. Technol. 2022; 14:3321–3331. doi: 10.1007/s41870-022-00864-6.

[18] Cherfly K, Yoga P. The Effect of Class Imbalance Handling on Datasets Toward Classification Algorithm Performance. *Matrik: jurnal manajemen, teknik informatika, dan rekayasa komputer*, 2023;22(2):227–238, doi: 10.30812/matrik.v22i2.2515

[19] Mounnan O, Manad O, Boubchir L, Mouatasim AE, Daachi B. A review on deep anomaly detection in blockchain. Blockchain Res. Appl., 2024;5(2):100187, doi: 10.1016/j.bcra.2023.100187.

[20] Chen J, Wang X, Lei F. Data-driven multinomial random forest: A new random forest variant with strong consistency. J. Big Data, 2024;11:34, doi: 10.1186/s40537-023-00874-6.

[21] Aliyev V. Ethereum Fraud Detection Dataset [Internet]. Kaggle [cited 2025 Feb 28]. Available from: https://www.kaggle.com/datasets/vagifa/ethereum-frauddetection-dataset