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], 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 - Since its inception, the pseudonymous and decentralised nature of Ethereum has made it challenging to trace fraud. The fraud type identified is the fraudulent reuse of the same funds from copied unique received addresses and a different time between the first and last user transactions. This study compares and examines 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 tend to overlook lightweight, interpretable models that may be utilised for real-time filtering or focus on Bitcoin rather than Ethereum. To fill this gap, this study utilises 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 was extracted from a publicly available repository called Ethereum Fraud Detection. The dataset was normalised 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 pre-processing. With 90.40% accuracy and an 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 visualisation of the confusion matrix, which also identifies areas for improvement in identifying actual fraud cases. These results provide business organisations with actionable recommendations on how to deploy real-time detection, rank high-risk transaction signals, and utilise adaptive machine learning architectures. This study promotes further research into anomaly detection based on deep learning and demonstrates the potential of feature-based machine learning in bolstering the 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. Introduction

The rapid growth of blockchain technology, particularly Ethereum, has revolutionised digital transactions. However, these innovative frameworks also present unique challenges, particularly in fraud detection. According to the official Ethereum website, Ethereum is a decentralised blockchain platform that enables developers to build and deploy smart contracts and decentralised applications [1]. Major cryptocurrencies, such as Bitcoin and Ethereum, experienced severe devaluation in 2022, with prices dropping sharply in a short period [2]. While Bitcoin primarily facilitates currency exchange, Ethereum was designed as a programmable platform for smart contracts and decentralised applications (dApps). Ethereum introduced a programmable blockchain framework that supports smart contract and decentralised application development. 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 strengthens the integrity and security of blockchain records by preventing unauthorised modifications. As a result, attackers cannot easily manipulate historical transaction data without controlling a substantial portion of the network's computational resources. These characteristics make Ethereum a secure platform for decentralised applications and for digital transactions. This flexibility has made Ethereum a cornerstone of blockchain innovation, enabling developers to build versatile, efficient solutions across industries.

Fraud detection on Ethereum poses unique challenges because of its decentralised 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 in seconds. According to the prosecution, they did so by exploiting a flaw in the MEV-boost software code, which the majority of Ethereum network "validators" use [3]-[4]. Traditional fraud detection methods, which are often designed for centralised 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 Markets in Crypto-Assets regulation, which takes effect in July 2024 and aims to standardise protections and market stability [5]. The platform's open nature allows malicious actors to exploit vulnerabilities and conduct fraudulent activities, such as unauthorised transactions or manipulation, without immediate detection. Identifying fraudulent transactions in this environment requires advanced analytical techniques that can process large volumes of data and uncover hidden patterns indicative of fraud. This study addresses the insufficiency of traditional fraud detection techniques in decentralised, high-dimensional blockchain networks such as Ethereum. Traditional approaches often lack scalability, interpretability, and the ability to act on fraudulent attempts in real time. Most existing studies focus on specific fraud types in artificial simulations rather than real transactional data. In decentralised systems, the absence of a central authority makes manual inspection or predefined, rule-based fraud detection infeasible. Therefore,

*CORRESPONDING AUTHOR | Nathania Vannesa | ✉ vannesanath@gmail.com
© 2026 The Author(s). Published by Universiti Malaysia Pahang Al-Sultan Abdullah Press. This is an open access article under the CC BY-NC 4.0 license

automatic intelligent systems that can examine large volumes of blockchain data, extract subtle patterns, and accurately classify fraudulent behaviour are in high demand.

This study makes several novel contributions to the blockchain fraud detection literature. The current work tends to address smart contract-based fraud or general cryptocurrency networks; this research explicitly addresses Ethereum transactional data. Second, while current studies point to complex models such as deep learning, which tend to sacrifice interpretability for performance, this study uses interpretable machine learning models that can directly explain how decisions are made. Third, using PySpark for training and testing ensures the method is not only scientifically feasible but also practical for implementation in real-time fraud detection systems, helping secure the 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 remainder of this paper is organised as follows: Section 2 reviews the relevant research on blockchain fraud detection, including its strengths, limitations, and relevance to this study. Section 3 describes the methodology, including data preprocessing, model implementation and performance metrics. Section 4 presents the results and compares the models' performance. Finally, Section 5 concludes the paper and outlines the future research directions.

2. Related Works

As reported by Elmougy and Manzi in the context of implementing Deep Learning for Ethereum Detection [6]. This study investigated classification algorithms to obtain the best-performing model with high precision and recall. The classification algorithms used in this study included J48, Random Forest, and Stochastic Gradient Descent. The authors focused on the precision and recall of Ponzi instances. The authors selected SSTORE, POP, MSTORE, and SWAP1 as features for an efficient classification model [7]. Statistical comparison showed that Random Forest achieved the highest recall and F-score among the evaluated models. However, SGD achieved 0.98 precision and 0.96 F-score. SGD achieved higher precision than the other methods. The authors demonstrated that the proposed model can effectively identify Ponzi schemes by flagging suspicious smart contracts during blockchain deployment. With all features, the model achieved a precision of 0.99 and a recall of 0.97. Koa et al. [8] proposed an Ethereum-based decentralised 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, thereby 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 study highlights blockchain's potential to decentralise trust in PKI systems, although real-world scalability remains unexplored. This study focuses on simulation-based evaluation across different node types. This study contributes to advancing innovative blockchain-based PKI solutions.

Table 1. Related works

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

Siddamsetti et al. [9] proposed a machine learning framework for detecting fraudulent transactions in Bitcoin and Ethereum networks. Using GPU-accelerated algorithms such as support vector machines, random forests, and logistic regression, the study analysed large-scale blockchain transaction metadata and achieved strong fraud detection performance. The models achieved 96.9% accuracy and 0.987 recall for Bitcoin, and 80.2% accuracy and 0.835 recall for Ethereum. Through sensitivity analysis, this study provides insights into key fraud indicators while demonstrating the potential for real-time anomaly detection. The methodology is adaptable beyond cryptocurrencies, offering applications for healthcare, government, and financial blockchain systems seeking to enhance security against malicious actors. This study advances blockchain integrity by combining technical innovation with empirical validation at scale. This study

addresses transactional fraud in Ethereum using interpretable machine learning models, namely Random Forest and Logistic Regression. This study uniquely focuses on Ethereum transactions and provides a comparative performance evaluation using a real-world dataset. The models were implemented in a PySpark environment that supports scalable real-time data processing. Beyond standard evaluation metrics, this study identifies the top predictive features that influence fraud detection, delivering both utility and interpretability. Table 1 explains the field of focus, strengths, and limitations. Compared with previous studies that focused on Ponzi schemes, smart contracts, or general cryptocurrency fraud, this study specifically focused on Ethereum transaction-level fraud detection using interpretable and scalable machine learning models.

3. Materials and Methods

This project leverages machine learning techniques to detect fraud on the Ethereum blockchain. The scope includes the following key aspects: Data Analysis and Preparation, Exploratory Data Analysis, Feature Importance Analysis, and Visualisation.

3.1 Metadata

The dataset was sourced from a publicly available repository titled 'Ethereum Fraud Detection' [10]. The data consisted of 9841 rows and 51 features; 39 columns were decimal, 9 were integers, and 3 were strings. Metadata, defined as structured information that describes, explains, or simplifies data retrieval and management [11], plays a critical role in ensuring the reproducibility and scalability of machine learning workflows, particularly in blockchain analytics [12]. The dataset consists of 9841 Ethereum transactions, of which approximately 1500 are labelled as fraudulent and 7000 as non-fraudulent. Because the dataset is imbalanced, future studies should apply class-balancing techniques, such as SMOTE, class weighting, or cost-sensitive learning, to improve fraud detection performance. This study focused on predicting transactions that indicate fraud; therefore, the features used were numeric. Prior studies have emphasised that blockchain metadata often requires preprocessing to handle high dimensionality and heterogeneity [13]. Table 2 presents the variables used.

Table 2. Features of the Ethereum fraud detection dataset

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 the account transaction 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 transactions.
5.	Sent tnx	The total quantity of regular transactions sent.
6.	Received tnx	The total quantity of regular transactions received.
7.	Number of Created Contracts	The total number of contract transactions 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 from which transactions were sent from an account.
10.	Min Value Received	Minimum value in Ether received.
11.	Max Value Received	The highest amount of Ether was ever received.
12.	Avg Val Received	The average amount of Ether that was ever received.
13.	Min Val Sent	The lowest amount of Ether was ever transmitted.
14.	Max Val Sent	This was the highest amount of ether ever transmitted.
15.	Avg Val Sent	The mean amount of ether was sent.
16.	Min Value Sent to Contract	The minimum amount of ether is sent to a contract.
17.	Max Val Sent to Contract	The highest amount of Ether that can be transmitted to a contract is
18.	Avg Value Sent to Contract	Average Ether value sent to the contract.

Note: Variable names were retained from the original Ethereum Fraud Detection dataset. In the variable names, “tnx” denotes transaction(s), “Avg” denotes average, “Min” denotes minimum, “Max” denotes maximum, “Val” denotes value, “Diff” denotes difference, and “Mins” denotes minutes.

3.2 Data Analysis and Preparation

This study implemented the model and feature scaling in PySpark, which enables real-time processing of large-scale data in a distributed environment using Python [14]. Table 3 shows the steps required to train and evaluate the models. We removed non-predictive identifiers and irrelevant ERC20-related fields to reduce noise and prevent overfitting. ERC20 is not valuable for this machine learning model because it focuses only 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 were non-predictive identifiers and did not contribute to fraud classification. Missing values were addressed by either mean imputation or removal. In this observation, *dropna* was implemented to remove missing values. After cleaning the data to remove missing values, dropping duplicates was also important. Table 4 presents the cleaned data. Normalisation is the process of organising a dataset to reduce redundancy and improve data integrity. We can normalise the MinMaxScaler using PySpark by importing *pyspark.ml.feature. MinMaxScaler.* Table 5 presents the results. The scaling process successfully normalised the raw transaction time metrics from the original scattered values into a uniform range between 0 and 1. This standardised presentation in Table 5 ensures that both features are systematically prepared for unbiased model training.

Table 3. Algorithm for Ethereum fraud detection

No	Step
1.	The required libraries on Pyspark, such as SparkSession, MinMaxScaler, and seaborn, are imported.
2.	Load the data into a variable
3.	The data were cleaned of missing 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 the multiclass classification evaluator.
12.	Confusion Matrix for both model
13.	Visualization top 10 important features

Table 4. Amount of clean data

Item	Status	Amount of data	
		Before	After
Missing value	No	9841	9841
Duplicate value	Yes	9841	9268

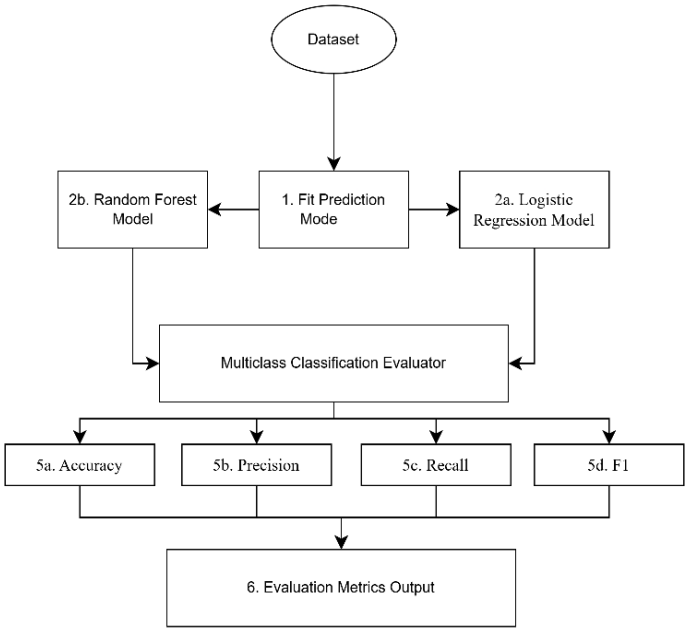


Figure 1. Flowchart of the Ethereum fraud detection process

The algorithm shown in Figure 1 outlines the steps required to fit the prediction model from the dataset using logistic regression (LR) and random forest (RF). This study used an evaluator called the Multiclass Classification Evaluator. We used four metrics: accuracy, precision, recall, and F1. Each serves a specific purpose. Accuracy indicates that the model

was correctly classified. Recall indicates how well the model captures most of the actual data. F1 measures the balanced ratio between the precision and recall.

Table 5. Scaled features of sent and received Ethereum transactions

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

Note: Variable names were retained from the original Ethereum Fraud Detection dataset. In the variable names, "tnx" denotes transaction(s), "Avg" denotes average, "Min" denotes minimum, "Max" denotes maximum, and "Min" denotes minutes.

3.3 Modeling

A comprehensive EDA was conducted to identify distribution patterns and feature correlations, providing valuable insights into the data structure. Exploratory Data Analysis is critical for understanding the underlying data characteristics and guiding model selection, particularly in high-dimensional financial datasets [15]. The project applies both logistic regression and random forest models to classify the transactions. These models were chosen because they handle structured data and provide interpretable results. Exploratory Data Analysis will cover correlation analysis, logistic regression, and random forests to predict fraudulent transactions on Ethereum. Logistic Regression was applied as a baseline model to predict the binary result of fraudulent transactions. Although simple, it provides initial insights into the data. The model produces a binary output representing fraudulent (1) and non-fraudulent (0) transactions. Logistic regression uses the logistic function to model the relationship between the predictor variables and the binary outcome. Logistic Regression is well suited to data mining challenges, including multicollinearity, missing values, and redundant features [16]. The logistic function is commonly derived from the formula in Eq. (1).

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

Eq. (1) represents the probability function of logistic regression; y_i is the dependent (response) variable. It represents the binary outcome for observation *i,* and x_i is the independent variable or feature vector for observation *i.* The last symbol is β , which represents a coefficient vector (parameter vector) associated with each feature in x_i. This model is robust to collinearity and missing data [17], making it suitable for preliminary fraud detection [18]. However, its linearity assumption limits its ability to capture complex nonlinear patterns inherent in blockchain transactions.

Random Forest, an ensemble learning method, suits both classification and regression tasks. As 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 imbalanced datasets better than Logistic Regression. Imbalanced data is a subtle but important problem that is easy to ignore until the model begins to prefer one class over another [19]. Random Forest has proven particularly effective in detecting subtle patterns in transaction timing [20]. The value of a random vector sampled independently and with the same distribution for every tree in the forest determines each tree's value in a random forest, which is a combination of the tree predictors [21]. For the Random Forest classifier, the model was set to use 20 trees and two output classes.

4. Results and Discussion

Correlation analysis is used to determine the relationship between features. Correlation values range from -1 to 1. A strong positive relationship is shown as 1, and in the Correlation Matrix, it appears red. The code produces a heatmap that shows the relationships between variables in the data. As shown in the correlation matrix in Figure 2, Sent transaction (Sent tnx) and Total transaction (including tnx to create contract) reach 0.7 out of 1, and Received Transaction and Total transaction reach 0.8 out of 1. Both sent and received transactions have a strong relationship with total transactions (including tnx to create a contract). Therefore, total transaction value depends heavily on sent and received transactions. Table 6 shows the accuracy, precision, recall, and F1 results for the logistic regression model. Based on the evaluation results, Logistic Regression achieved an accuracy of 81.04%, indicating that the model correctly classified most transactions. A 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 actual data. The F1 score of 73% reflects a balanced trade-off between precision and recall. A confusion matrix visualises the distribution of fraud predictions.

Table 6. Performance of Logistic Regression

Metrics	Results (%)
Accuracy	81.04
Precision	66.42
Recall	81.04
F1	73.00

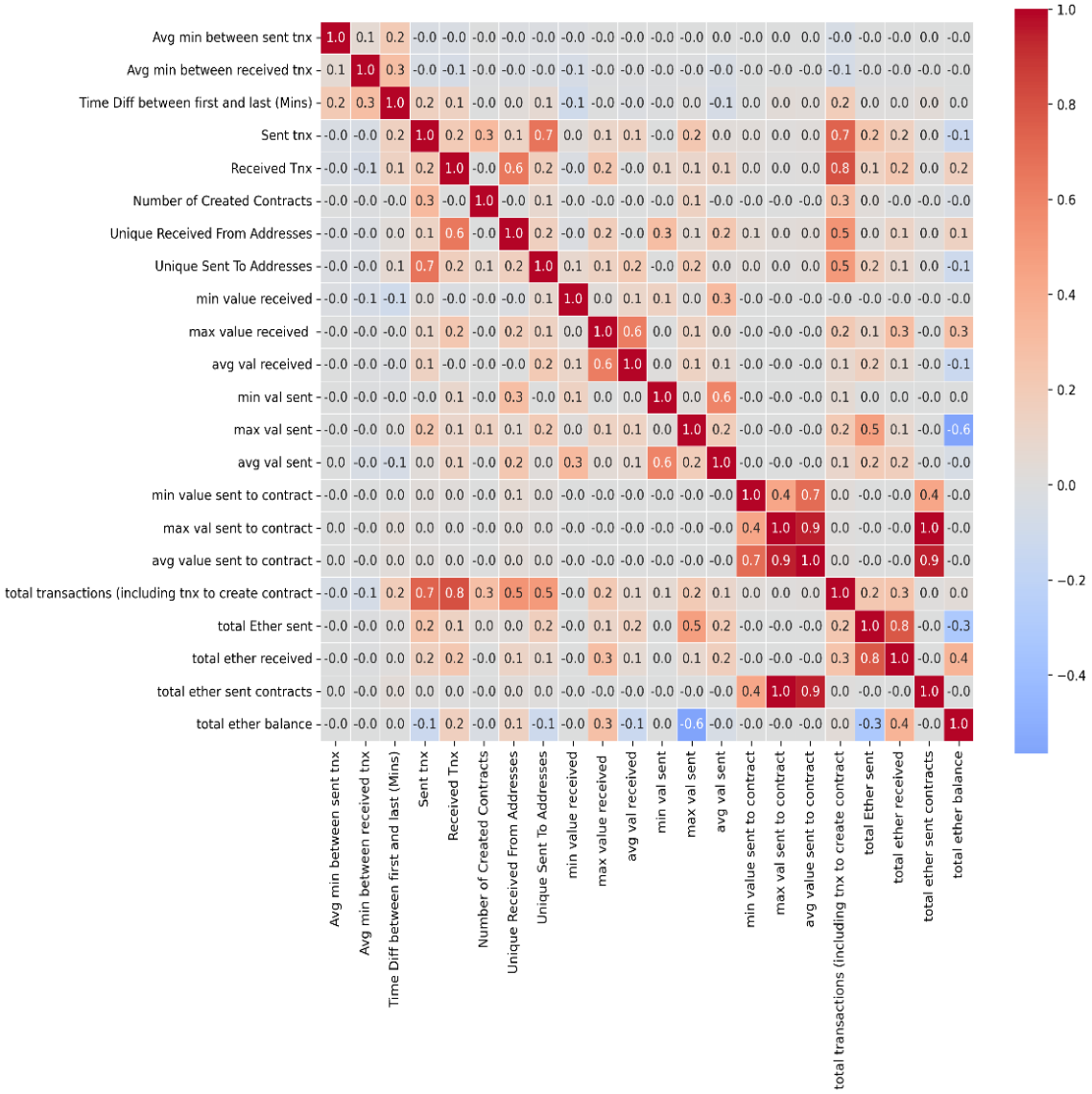


Figure 2. Correlation matrix of features

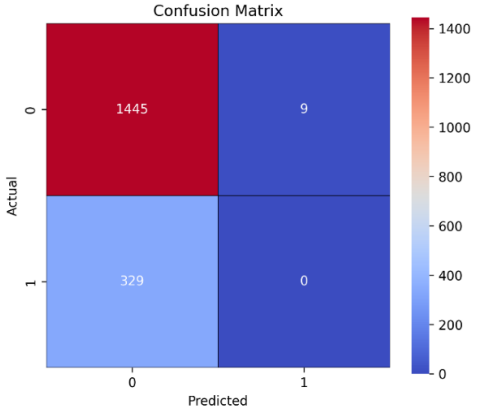


Figure 3. Confusion matrix of Logistic Regression

The confusion matrix for Logistic Regression, as shown in Figure 3, summarises the predicted classifications against the actual classes. Logistic Regression performed strongly at identifying non-fraudulent transactions, producing 1,445 true negatives and only nine false positives. However, Logistic Regression performed poorly in identifying fraudulent transactions, producing 329 false negatives and no true-positive detections. Table 7 shows the accuracy, precision, recall, and F1 results for the Random Forest model. Based on the evaluation results, Random Forest achieved 90.40% accuracy, indicating that the model correctly classified most transactions in the dataset. The precision of 90.89% indicates that most transactions classified as fraudulent were accurately 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 generalise was also verified by comparing training and test accuracy. The training accuracy was 91%, indicating that the Random Forest model was not overfitted and maintained high prediction consistency on unseen data. A confusion matrix was employed to visualise the distribution of fraud predictions.

Table 7. Performance of Random Forest

Metrics	Results (%)
Accuracy	90.40
Precision	90.89
Recall	90.40
F1	89.17

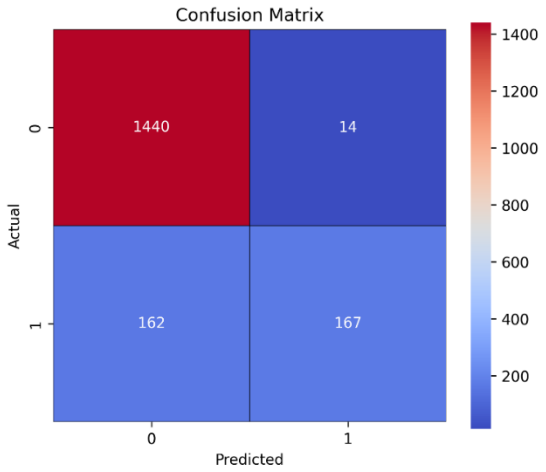


Figure 4. Confusion matrix of Random Forest

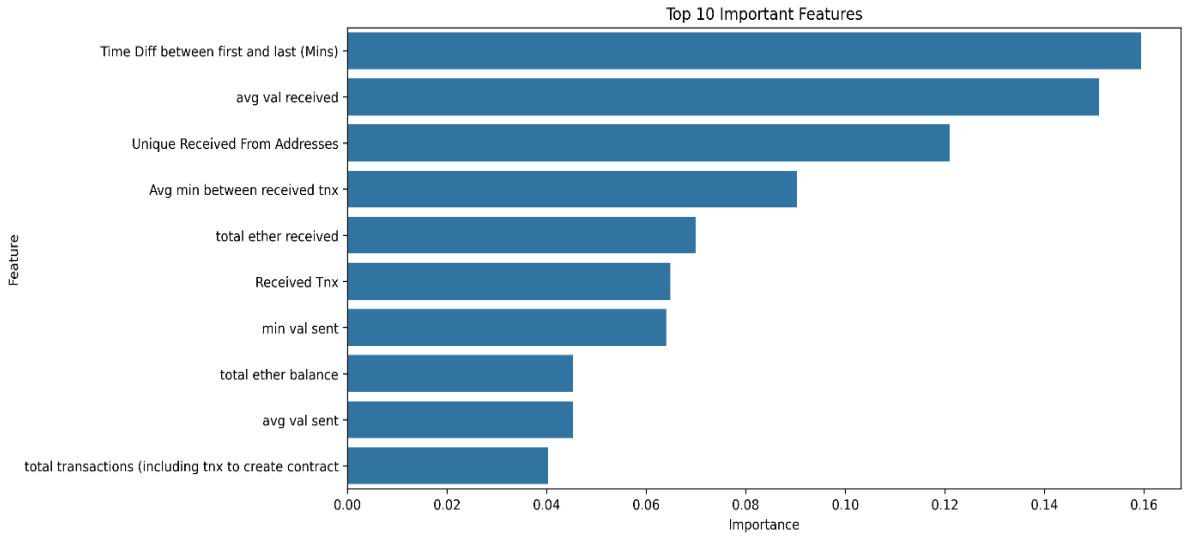


Figure 5. Top 10 important features

The confusion matrix for the Random Forest in Figure 4 summarises the predicted classifications against the actual classes. The Random Forest model 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 positives and 162 false negatives. Although Random Forest achieved high overall accuracy, the number of false negatives indicated

that further optimisation was needed to improve fraud-detection sensitivity. This project emphasises identifying key features that significantly influence fraud detection, offering practical insights into the behaviour of fraudulent transactions.

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', and 'minimum value received, which can guide fraud detection by prioritising these metrics for monitoring. The top 10 attributes identified as crucial for Ethereum fraud detection show strong behavioural and transactional patterns connected to fraudulent operations, including the unique value received from the address, time diff between the first and last transactions, total Ether sent/received, and average/min values. According to these findings, companies should prioritise real-time monitoring systems to identify irregularities, such as abrupt increases in Ether volume or rapid transactions involving multiple distinct addresses. Companies can automate risk scoring by implementing a Random Forest model with 90% accuracy. 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 standardise fraud protection and balance security with usability. By focusing on these three characteristics, businesses can develop a proactive, flexible approach that improves detection rates, reduces false positives, and increases confidence in Ethereum-based transactions. To counter new fraud strategies, future work should explore deep learning for subtle pattern recognition and dynamic model retraining. The best model for Ethereum Fraud detection is a Random Forest with 90% accuracy and 89% F1 score. Unique Received From Address, Time Diff between first and last (Mins), average Val received, total Ether sent, total Ether received, average minimum between received transactions, minimum Val sent, total transactions, received transactions, and minimum value received are significant predictors of fraud. Specific patterns and behaviours, such as the number of unique addresses involved, strongly influence suspicious financial transactions, as shown in Table 8. This observation highlights the critical need for companies using Ethereum for their transactions to implement robust monitoring systems to detect and prevent fraud.

Table 8. Evaluation of Random Forest

Feature	Risk indicator	Mitigation strategy
Unique Received Address	A high number of unique received addresses may indicate potential money laundering.	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

5. Conclusions

This study demonstrated the potential of machine learning to detect fraudulent transactions on the Ethereum blockchain. Logistic Regression and Random Forest were the primary classification methods, and these models made significant progress in identifying patterns associated with fraud. With 90.4% accuracy and an F1-score of 89.17%, the results suggest that Random Forest performs better than Logistic Regression. Although Random Forest outperformed Logistic Regression, the model still produced false negatives in fraud detection. Therefore, future studies should explore class balancing, anomaly detection, deep learning, and real-time adaptive model retraining. In addition, ensemble and deep learning approaches may further improve fraud detection sensitivity while maintaining acceptable interpretability. The insights and models developed in this study provide a foundation for more secure and trustworthy Ethereum transactions. The findings also revealed that specific behavioural patterns, such as the number of unique addresses involved and time gaps between transactions, play a significant role in identifying suspicious activities. For businesses, these findings emphasise the importance of focusing on fraud detection efforts in high-risk categories and leveraging key transactional features to flag anomalies in real time. By doing so, companies can safeguard their platforms against fraudulent activities and ensure a secure and trustworthy environment for their customers. This proactive approach enhances customer confidence, making them feel safer and more comfortable conducting transactions, ultimately fostering customer loyalty and supporting business growth.

Acknowledgements

The authors thank the Faculty of Computing, Universitas Pembangunan Nasional "Veteran", Surabaya, Indonesia, for providing the infrastructure needed to carry out this research.

Funding

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

Declaration of Competing Interests

The authors declare no conflict of interest.

Credit Authorship Contribution Statement
Nathania Vannesa: Conceptualisation, Methodology, Data curation, Writing – original draft
Nusrat Zahan Nisha: Analysis; Writing – review & revise; Project administration
Shindi Shella May Wara: Investigation; Validation; Supervision; Writing – review & revise

Availability of Data and Materials
Data sharing is not applicable to this article, as no new datasets were generated or analysed in this study.

Ethics Declarations
This study did not involve human participants or animal subjects. Therefore, ethical approval was not required.

Generative Artificial Intelligence Declarations
No generative artificial intelligence tools were used to prepare this manuscript. All intellectual content is original and is solely attributable to the authors.

References
[1] Ethereum.org, "What is Ethereum?," 2025. [Online]. Available: https://ethereum.org/en/what-is-ethereum/. Accessed: Feb. 28, 2025.
[2] Q. Huang, "Ethereum: Introduction, expectation, and implementation," *Highlights in Science, Engineering and Technology*, vol. 39, pp. 1–7, 2023, https://doi.org/10.54097/hset.v39i.6188
[3] A. Belanger, "MIT students stole $25M in seconds by exploiting ETH blockchain bug, DOJ says," *Ars Technica*, May 16, 2024. [Online]. Available: https://arstechnica.com/tech-policy/2024/05/sophisticated-25m-ethereum-heist-took-about-12-seconds-doj-says/. Accessed: Feb. 28, 2025.
[4] C. Isidore, "Two former MIT students charged with stealing $25 million of crypto in 12 seconds," *CNN Business*, May 16, 2024. [Online]. Available: https://edition.cnn.com/2024/05/16/investing/mit-crypto-hack/index.html. Accessed: Feb. 28, 2025.
[5] European Parliament and Council of the European Union, "Regulation (EU) 2023/1114 of the European Parliament and of the Council on markets in crypto-assets (MiCA)," *Official Journal of the European Union*, 2023. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32023R1114. Accessed: Feb. 28, 2025.
[6] Y. Elmougy and O. Manzi, "Anomaly detection on Bitcoin, Ethereum networks using GPU-accelerated machine learning methods," in Proceedings of the 31st International Conference on Computer Theory and Applications (ICCTA), Alexandria, Egypt, 2021, pp. 166–171, https://doi.org/10.1109/ICCTA54562.2021.9916625
[7] I. J. Onu, A. E. Omolara, M. Alawida, O. I. Abiodun, and A. Alabdultif, "Detection of Ponzi scheme on Ethereum using machine learning algorithms," *Scientific Reports*, vol. 13, Art. no. 18403, 2023, https://doi.org/10.1038/s41598-023-45275-0
[8] C. G. Koa, S. H. Heng, and J. J. Chin, "New Ethereum-based distributed PKI with a reward-and-punishment mechanism," *Blockchain: Research and Applications*, vol. 6, no. 1, Art. no. 100239, 2025, https://doi.org/10.1016/j.bcra.2024.100239
[9] S. Siddamsetti and M. Srivenkatesh, "Efficient fraud detection in Ethereum blockchain through machine learning and deep learning approaches," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 11, no. 11s, pp. 71–82, 2023, https://doi.org/10.17762/ijritcc.v11i11s.8072
[10] V. Aliyev, "Ethereum fraud detection dataset," Kaggle, 2021. [Online]. Available: https://www.kaggle.com/datasets/vagifa/ethereum-frauddetection-dataset. Accessed: Feb. 28, 2025.
[11] M. L. Zeng and J. Qin, *Metadata*, 3rd ed. Chicago, IL, USA: ALA Neal-Schuman, 2022. ISBN: 978-0-8389-4875-5.
[12] T. Ashfaq, R. Khalid, A. S. Yahaya, *et al.*, "A machine learning and blockchain based efficient fraud detection mechanism," *Sensors*, vol. 22, no. 19, Art. no. 7162, 2022, https://doi.org/10.3390/s22197162
[13] P. Azad, C. G. Akcora, and A. Khan, "Machine learning for blockchain data analysis: Progress and opportunities," *Distributed Ledger Technologies: Research and Practice*, vol. 5, no. 1, Art. no. 10, 2026, https://doi.org/10.1145/3728474
[14] Apache Software Foundation, "PySpark overview," Apache Spark. [Online]. Available: https://spark.apache.org/docs/latest/api/python/index.html. Accessed: Feb. 28, 2025.
[15] I. G. T. Isa, Z. Zulkarnaini, L. Novianti, F. Elfaladonna, and S. Agustri, "Exploratory data analysis (EDA) dalam dataset penerimaan mahasiswa baru Universitas XYZ Palembang" [Exploratory data analysis (EDA) on the new-student admission dataset of University XYZ Palembang], *Smart Comp: Jurnalnya Orang Pintar Komputer*, vol. 12, no. 3, pp. 600–609, 2023, https://doi.org/10.30591/smartcomp.v12i3.4125
[16] S. Jiang, J. Josse, and M. Lavielle, "Logistic regression with missing covariates—Parameter estimation, model selection and prediction within a joint-modeling framework," *Computational Statistics & Data Analysis*, vol. 145, Art. no. 106907, 2020, https://doi.org/10.1016/j.csda.2019.106907
[17] Z. Gu and O. Dib, "Enhancing fraud detection in the Ethereum blockchain using ensemble learning," *PeerJ Computer Science*, vol. 11, Art. no. e2716, 2025, https://doi.org/10.7717/peerj-cs.2716

[18] R. M. Aziz, M. F. Baluch, S. Patel, and A. H. Ganie, "LGBM: A machine learning approach for Ethereum fraud detection," *International Journal of Information Technology*, vol. 14, no. 7, pp. 3321–3331, 2022, https://doi.org/10.1007/s41870-022-00864-6

[19] C. Kaope and Y. Pristyanto, "The effect of class imbalance handling on datasets toward classification algorithm performance," *MATRIK: Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, vol. 22, no. 2, pp. 227–238, 2023, https://doi.org/10.30812/matrik.v22i2.2515

[20] O. Mounnan, O. Manad, L. Boubchir, A. El Mouatasim, and B. Daachi, "A review on deep anomaly detection in blockchain," *Blockchain: Research and Applications*, vol. 5, no. 4, Art. no. 100227, 2024, https://doi.org/10.1016/j.bcra.2024.100227

[21] J. Chen, X. Wang, and F. Lei, "Data-driven multinomial random forest: A new random forest variant with strong consistency," *Journal of Big Data*, vol. 11, Art. no. 34, 2024, https://doi.org/10.1186/s40537-023-00874-6