http://dx.doi.org/10.21622/ACE.2026.06.1.1878

ETHEREUM FRAUD DETECTION USING SMOTE-TOMEK AND STACKED ENSEMBLE LEARNING TECHNIQUE

Jamiu Muhammed Usman[1], Muhammad Nazeer Musa[2], Mario Kolberg[3], Nafisat Abdulkadir[4] and Habib Mohammed[5]

[1]University of Stirling, United Kingdom

[2]NIGERIAN DEFENCE ACADEMY, KADUNA, NIGERIA

[3]University of Stirling, United Kingdom

[4]Confluence University of Science and Technology, Osara, Nigeria

[5]North Carolina State University, Raleigh, USA

Emails: {jmu00001@students.stir.ac.uk, muhammadmusa2502@nda.edu.ng, mario.kolberg@stir.ac.uk, abdulkadirn@custech.edu.ng, himohamm@ncsu.edu}

Received 29 December 2025 - Accepted 02 February 2026 - Published 26 April 2026

ABSTRACT

This research addresses the critical challenge of detecting fraudulent Ethereum transactions, which remains notoriously difficult due to the overwhelming prevalence of legitimate activities compared to illicit ones. This pronounced class disparity frequently results in skewed outcomes when employing conventional detection frameworks. Our investigation implements a composite resampling strategy, SMOTE-Tomek, which concurrently augments the under-represented fraud category whilst eliminating ambiguous instances from the dominant category. Subsequently, the equalised dataset undergoes processing via a stacked ensemble framework comprising three robust gradient boosting methodologies: XGBoost, LightGBM, and CatBoost, synthesised by a logistic regression meta-classifier. Through stratified 5-fold cross-validation conducted against established benchmarks, the proposed ensemble attained a mean accuracy of 0.9873, whilst precision, recall, and F1-score each achieved an exceptional score of 0.98. Benchmark comparisons verified that our stacked methodology consistently surpasses individual base algorithms. The study demonstrated that combining targeted resampling with ensemble stacking offers highly effective solutions for Ethereum fraud detection, though real-time implementation and cross-blockchain generalisability require further investigation. The proposed model provides cryptocurrency exchanges, investors, and regulatory bodies with a robust mechanism for identifying fraudulent activities, thereby enhancing ecosystem security and maintaining user confidence. This work represents the first comprehensive integration of SMOTE-Tomek resampling with stacked gradient boosting ensembles for Ethereum fraud detection, validated through rigorous k-fold cross-validation rather than simple train-test splits.

Keywords: *Ethereum Fraud Detection, XGBoost, LightGBM, Stacking Ensemble, SMOTE-Tomek.*

1. INTRODUCTION

Since its establishment by Vitalik Buterin in 2015, Ethereum has distinguished itself as a prominent blockchain platform, largely attributed to its groundbreaking deployment of smart contracts and autonomous agreements with predetermined conditions encoded directly [1]. These programmable contracts have fundamentally transformed digital commerce and interactions. Ethereum's decentralised infrastructure, coupled with its Turing-complete

http://dx.doi.org/10.21622/ACE.2026.06.1.1878

programming capabilities, has enabled the proliferation of diverse decentralised applications (DApps) spanning numerous sectors, including decentralised finance (DeFi), gaming, and supply chain logistics. This proliferation has resulted in substantial total value locked (TVL) within Ethereum's smart contract ecosystem [2]. Nevertheless, the accelerated expansion and mounting complexity of DApps have simultaneously introduced potential security vulnerabilities subject to exploitation [3]. Analogous to traditional financial systems, Ethereum remains vulnerable to fraudulent schemes, whilst its decentralised architecture presents distinctive security obstacles [4]. Traditional centralised security protocols frequently prove inadequate within Ethereum's operational environment, contributing to escalating fraudulent incidents that compromise both individual users and the broader cryptocurrency ecosystem [4].

Multiple fraudulent schemes, including Ponzi operations masquerading as legitimate DeFi initiatives, have become increasingly commonplace on the Ethereum platform. These fraudulent ventures promise unrealistic returns and utilise capital from subsequent investors to remunerate earlier participants, thereby maintaining an untenable cycle [2]. According to the 2025 Crypto Crime Report by Chainalysis, the estimated value of illicit cryptocurrency transactions in 2024 reached USD 40.9 billion [5]. Furthermore, phishing campaigns exploit the anonymity and restricted recourse mechanisms available to users, presenting considerable threats to private key security and confidential information [6]. Reentrancy vulnerabilities, which exploit weaknesses in smart contract architecture, have occasioned substantial financial damages, underscoring the imperative for secure programming methodologies [7]. These fraudulent operations have inflicted significant monetary losses, eroding confidence in the Ethereum ecosystem and impeding its widespread adoption [4].

Robust fraud detection and prevention mechanisms are therefore essential. Conventional approaches, such as manual auditing procedures, prove insufficient to address the evolving character of Ethereum-based fraud [4]. Machine learning, with its pattern recognition capabilities, has emerged as a promising alternative. Previous investigations have examined algorithms including Support Vector Machines (SVM), Random Forests (RF), and neural networks for Ethereum fraud detection, demonstrating their potential for enhanced accuracy and efficiency [8], [9]. Recent advances have introduced novel approaches, including transaction language models combined with graph representation learning [10], ensemble learning frameworks with blockchain [11], and temporally validated detection systems that address the evolution of phishing behaviours to provide more realistic performance estimates [12]. Nevertheless, persistent challenges remain, particularly imbalanced datasets wherein fraudulent transactions occur significantly less frequently than legitimate ones, potentially introducing model bias [4]. The dynamic character of fraud likewise necessitates adaptable models.

Given the increasing cryptocurrency adoption and substantial financial losses attributable to fraud [4], effective detection mechanisms are vital for maintaining confidence in the Ethereum ecosystem. Such mechanisms benefit investors, exchanges, and regulatory bodies by facilitating more secure transactions. This investigation seeks to develop an advanced fraud detection model for Ethereum utilising stacked ensemble learning whilst examining the efficacy of data balancing strategies, specifically SMOTE-Tomek and SMOTE-ENN.

2. OVERVIEW OF ETHEREUM AND ITS SECURITY CHALLENGES

2.1 ETHEREUM ARCHITECTURE

Ethereum, frequently characterised as "the world computer," represents a decentralised, open-source blockchain platform distinguished by its innovative utilisation of smart contracts [1]. These autonomous agreements, with terms directly encoded in programming code, facilitate trustless transactions and interactions. Ethereum's architecture employs distributed ledger technology, wherein transactions are aggregated into blocks and appended to the blockchain through a consensus mechanism, ensuring data immutability and transparency. The Ethereum Virtual Machine (EVM) constitutes the platform's core,

http://dx.doi.org/10.21622/ACE.2026.06.1.1878

functioning as a runtime environment for executing smart contract bytecode [13]. The EVM is engineered to be deterministic and isolated, ensuring predictable contract execution without interference from other contracts or the underlying blockchain infrastructure. The Ethereum network architecture is illustrated in Fig. 1.

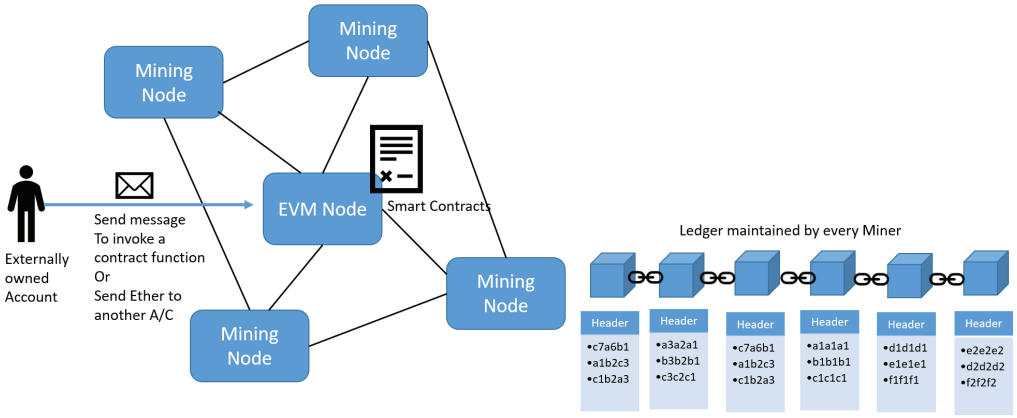


Figure 1: Ethereum Network Architecture

Fig. 1 depicts the fundamental components of the Ethereum network pertinent to fraud detection research. Fraudulent activities may originate from multiple sources, including Externally Owned Accounts (EOAs), which can be compromised for phishing operations or deployment of malicious smart contracts, thereby emphasising the significance of monitoring EOA behaviour. Additionally, Smart Contracts may be deliberately constructed for theft or data manipulation, necessitating a comprehensive analysis of both their source code and operational behaviour. Furthermore, examination of EVM Nodes can reveal patterns indicative of fraudulent activities. The data extracted from these components, as illustrated in Fig. 1, are employed to train machine learning models aimed at identifying fraudulent transactions and smart contract behaviours. The Ethereum network also encompasses Mining Nodes, whose monitoring is crucial for detecting potential network attacks, alongside the Ledger (Blockchain), which provides a tamper-resistant record of transactions that can illuminate historical fraudulent activities and inform the development of enhanced detection models. This underscores the necessity for a comprehensive approach to fraud detection within the Ethereum network, integrating various components and data sources.

2.2 REVIEW OF EXISTING RESEARCH ON ETHEREUM FRAUD DETECTION

Initial research on fraud detection within the Ethereum ecosystem predominantly utilised conventional methodologies, including rule-based systems, anomaly detection, and statistical analysis [14], [15]. Conversely, more recent investigations have transitioned towards the application of various machine learning techniques [3], [9], [16], [17], [18], [19], [20], [21], [22], [23], [24]. For instance, Taher et al. [22] introduced an ensemble learning model integrating Explainable AI, utilising a voting-based strategy incorporating Decision Tree (DT), RF, and AdaBoost classifiers. By employing Random Oversampling (ROS) and the Synthetic Minority Over-sampling Technique (SMOTE) for data balancing, this approach achieved 98% accuracy with soft voting (4.59 seconds training time) and 99% with hard voting. Feature selection was performed using LIME, resulting in a final set of 43 features after excluding those with missing values and high collinearity. However, reliance on a simple 80/20 train-test split without cross-validation raises concerns regarding potential overfitting, particularly in the context of data imbalance.

Md et al. [16] introduced a stacking classifier integrating multiple algorithms into a meta-learner, including Logistic Regression, Naive Bayes, DT, RF, AdaBoost, K-Nearest Neighbours (KNN), SVM, and Gradient Boosting. This stacking classifier utilised Multinomial Naive Bayes and RF as base learners, with logistic regression serving as the meta-learner. It achieved 97.18% accuracy and an F1-score of 97.02%. However, the study did not explicitly detail the

http://dx.doi.org/10.21622/ACE.2026.06.1.1878

data balancing techniques employed and relied solely on a train-test split for evaluation.

Ravindranath et al. [18] examined the effects of oversampling techniques on Ethereum fraud detection, utilising SMOTENC, ADA-SYN, and K-Means-SMOTE to address class imbalance. The ensemble models, particularly CatBoost and LightGBM, demonstrated significant efficiency, achieving accuracy rates between 98% and 98.54% after applying SMOTE oversampling. The findings revealed improved Performance with SMOTE compared to scenarios without oversampling (e.g., CatBoost accuracy/F1-score: 98.54% with SMOTE versus 98.24%/93.98% without). However, the use of train-test splitting limits the generalisability of the results. Feature selection involved the elimination of highly correlated and low-variance features.

Sallam et al. [9] employed KNN, RF, and XGBoost in combination with an ensemble weighted method for data balancing on a dataset consisting of 4,681 instances. This approach achieved average accuracies of 96.80% for XGBoost, 94.88% for RF, and 87.85% for KNN, using 42 extracted and cleansed features. A 90/10 train-test split was employed, with XGBoost demonstrating the highest accuracy.

Aziz et al. [3] employed XGBoost for Ethereum fraud detection, achieving 97.76% accuracy by applying SMOTE for data balancing. Feature extraction yielded a set of 20 features after eliminating highly correlated and low-variance features. However, the simplistic train-test split raises concerns about overfitting, indicating that alternative data balancing techniques should be explored.

Finally, Kabla et al. [17] proposed Eth-PSD, a machine learning-based approach for phishing scam detection in Ethereum with KNN optimisation. Addressing data imbalance with random oversampling, they achieved 98.11% accuracy with a low false positive rate (0.01) using cross-validation. However, the study focused solely on phishing scams and used only four key features.

Recent advancements have introduced innovative methodologies. Sun et al. [10] proposed TLMG4Eth, integrating transaction language models with graph-based methods to capture semantic, similarity, and structural features, achieving superior detection performance by addressing the limitation of unclear transaction semantics in previous approaches. Gu and Dib [11] presented an ensemble learning framework, demonstrating that Random Forest and XGBoost models effectively identify fraudulent transactions with high precision whilst maintaining computational efficiency for real-time detection. Ertam [12] introduced a temporally validated evaluation mechanism to address methodological gaps in previous studies. Their work demonstrated that earlier approaches, which relied on random train-test partitions, frequently overestimated real-world Performance by disregarding the temporal progression of phishing behaviours.

Despite advancements in Ethereum fraud detection, several research gaps persist. Whilst existing studies have demonstrated the potential of machine learning and ensemble methods, addressing data imbalance remains a challenge. Advanced data balancing techniques like SMOTE-Tomek and SMOTE-ENN, which mitigate noise and class overlap, are underutilised. Furthermore, the potential of stacked ensemble learning, which combines the strengths of multiple ensemble models, has not been fully explored in this domain. More vigorous evaluation methodologies, such as k-fold cross-validation, are needed to ensure model generalisability. Addressing these gaps is crucial for developing more accurate and reliable fraud detection models for the Ethereum blockchain.

3. RESEARCH METHODOLOGY

Our investigative methodology, depicted in Fig. 2, proceeds through four sequential phases: initial data acquisition, comprehensive data preparation and cleansing, construction of classification models leveraging supervised and unsupervised learning paradigms, and rigorous performance assessment of developed models.

http://dx.doi.org/10.21622/ACE.2026.06.1.1878

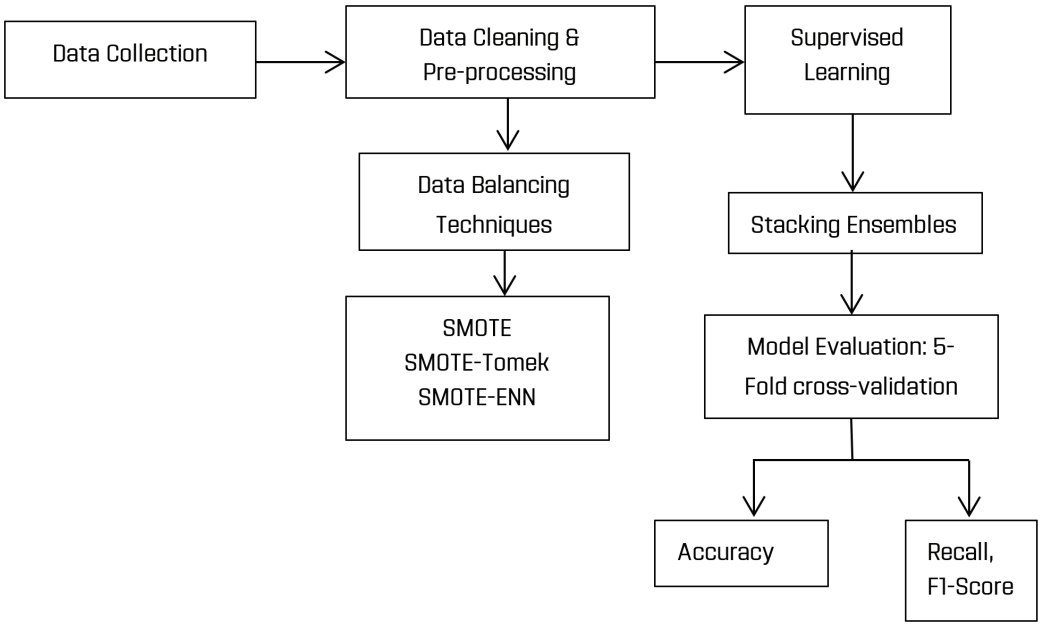


Figure 2: Research Methodology

3.1 DATA COLLECTION

Our empirical investigation leverages publicly available Ethereum transaction records compiled by Aliyev [25], representing a comprehensive repository of account-level blockchain activities pertinent to fraudulent behaviour identification. This repository encompasses multifaceted characteristics spanning account identifiers, transactional sequences, and economic indicators. The compilation captures quantitative measurements across multiple dimensions—transaction frequency, temporal patterns, monetary flows, and smart contract engagement—collectively constructing detailed financial profiles for individual accounts. As visualised in Fig. 3, this comprehensive data structure establishes the empirical foundation for behavioural pattern analysis and predictive model construction. The repository delivers granular visibility into account operations encompassing native Ether transfers alongside ERC20 token movements. Critical economic measurements embedded within include: cumulative transaction tallies, bidirectional Ether transfer volumes (dispatched and received), and terminal account balances. For token-specific activities, the data structure captures: ERC20 transaction frequencies, aggregate token-denominated value flows (both outbound and inbound), and unique smart contract interaction counts.

http://dx.doi.org/10.21622/ACE.2026.06.1.1878

Index	Column Name	Non-Null Count	Dtype
0	FLAG	9841 non-null	int64
1	Avg min between sent tnx	9841 non-null	float64
2	Avg min between received tnx	9841 non-null	float64
3	Time Diff between first and last (Mins)	9841 non-null	float64
4	Sent tnx	9841 non-null	int64
5	Received Tnx	9841 non-null	int64
6	Number of Created Contracts	9841 non-null	int64
7	Unique Received From Addresses	9841 non-null	int64
8	Unique Sent To Addresses	9841 non-null	int64
9	min value received	9841 non-null	float64
10	max value received	9841 non-null	float64
11	avg val received	9841 non-null	float64
12	min val sent	9841 non-null	float64
13	max val sent	9841 non-null	float64
14	avg val sent	9841 non-null	float64
15	min value sent to contract	9841 non-null	float64
16	max val sent to contract	9841 non-null	float64
17	avg value sent to contract	9841 non-null	float64
18	total transactions (including contract)	9841 non-null	int64
19	total Ether sent	9841 non-null	float64
20	total ether received	9841 non-null	float64
21	total ether sent contracts	9841 non-null	float64
22	total ether balance	9841 non-null	float64
23	Total ERC20 tnxs	9012 non-null	float64
24	ERC20 total Ether received	9012 non-null	float64
25	ERC20 total ether sent	9012 non-null	float64
26	ERC20 total Ether sent contract	9012 non-null	float64
27	ERC20 uniq sent addr	9012 non-null	float64
28	ERC20 uniq rec addr	9012 non-null	float64
29	ERC20 uniq sent addr.1	9012 non-null	float64
30	ERC20 uniq rec contract addr	9012 non-null	float64
31	ERC20 avg time between sent tnx	9012 non-null	float64
32	ERC20 avg time between rec tnx	9012 non-null	float64
33	ERC20 avg time between rec 2 tnx	9012 non-null	float64
34	ERC20 avg time between contract tnx	9012 non-null	float64
35	ERC20 min val rec	9012 non-null	float64
36	ERC20 max val rec	9012 non-null	float64
37	ERC20 avg val rec	9012 non-null	float64
38	ERC20 min val sent	9012 non-null	float64
39	ERC20 max val sent	9012 non-null	float64
40	ERC20 avg val sent	9012 non-null	float64
41	ERC20 min val sent contract	9012 non-null	float64
42	ERC20 max val sent contract	9012 non-null	float64
43	ERC20 avg val sent contract	9012 non-null	float64
44	ERC20 uniq sent token name	9012 non-null	float64
45	ERC20 uniq rec token name	9012 non-null	float64
46	ERC20 most sent token type	7144 non-null	object
47	ERC20_most_rec_token_type	8970 non-null	object

Figure 3: Variables in the Transaction Dataset

3.2 DATA PREPARATION

Data preparation represents the foundational phase wherein we establish dataset quality and coherence, directly influencing subsequent analytical validity. Within machine learning contexts, this phase proves particularly vital given that datasets commonly harbour inconsistencies, excessive correlations, or absent values. Our preparation workflow unfolds across multiple subsections, detailed below.

3.2.1 Exploratory Data Analysis

Beginning with 9,841 observations distributed across 48 attributes, we conducted preliminary reconnaissance. This exploratory phase involved scrutinising variable types and completeness metrics, deriving descriptive statistics, and constructing visual representations of target variable distributions through pie chart visualisation, as demonstrated in Fig. 4, which demonstrated the pronounced class disparity within our dataset: legitimate transactions constitute approximately 77.86% of observations, whilst fraudulent cases account for the remaining 22.14%.

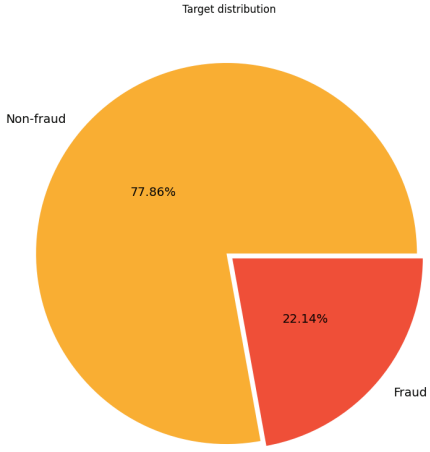


Figure 4: Target Distribution

http://dx.doi.org/10.21622/ACE.2026.06.1.1878

3.2.2 Handling Missing Values

We employed heatmap visualisation (Fig. 5) to identify patterns of missing observations. Our resolution strategy involved median imputation for numeric attributes, thereby preserving the complete observation set whilst addressing incomplete records.

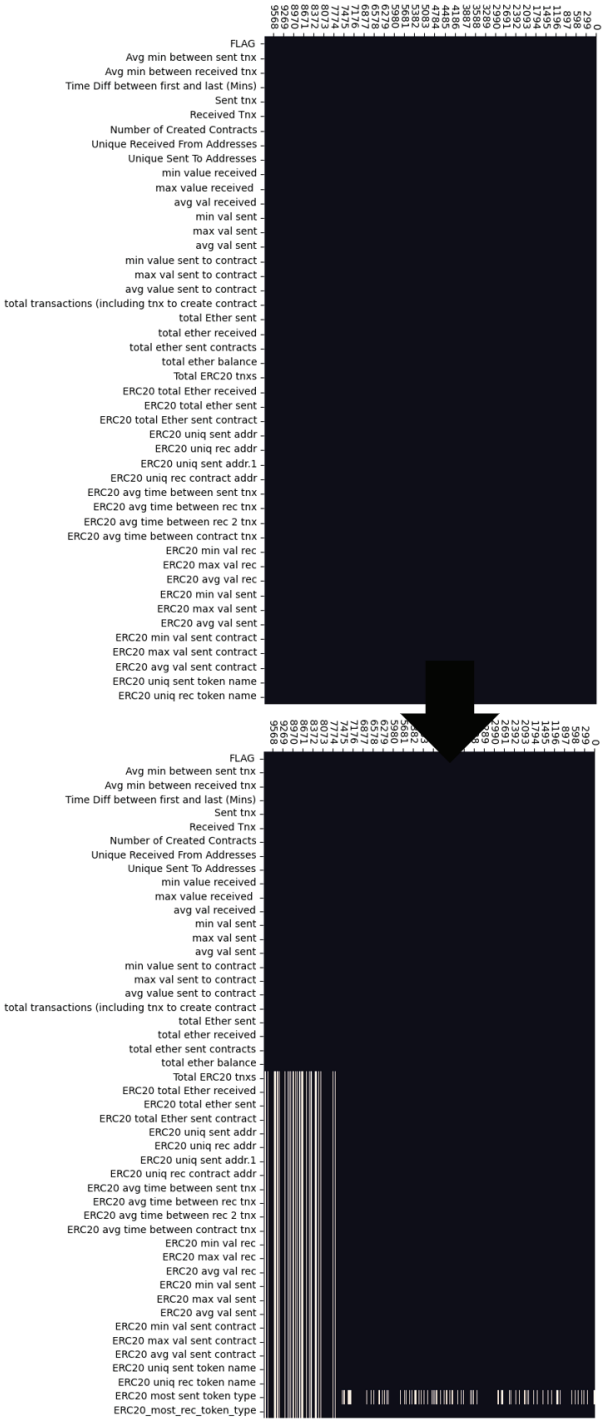


Figure 5: Visualization of Missing Values and Correction

3.2.3 Removing Zero-Variance Features

Attributes demonstrating zero variance contribute negligibly to predictive capacity and were consequently removed. This elimination encompassed seven ERC20-related temporal and contractual value attributes: 'ERC20 avg time between sent tnx', 'ERC20 avg time between rec tnx', 'ERC20 avg time between rec 2 tnx', 'ERC20 avg time between contract tnx', 'ERC20 min val sent contract', 'ERC20 max val sent contract', and 'ERC20 avg val sent contract'. This

http://dx.doi.org/10.21622/ACE.2026.06.1.1878

streamlining enhanced computational efficiency. Additionally, two categorical attributes—'ERC20 most sent token type' and 'ERC20_most_rec_token_type'—exhibiting excessive cardinality (305 and 467 distinct values, respectively) were excised to prevent overfitting and curtail computational burden.

3.2.4 Removing Highly Correlated Features

Through correlation matrix examination, we identified and eliminated attributes exhibiting strong intercorrelation, thereby addressing multicollinearity, a phenomenon known to compromise both model interpretability and predictive efficacy. The refined correlation structure appears in Fig. 6.

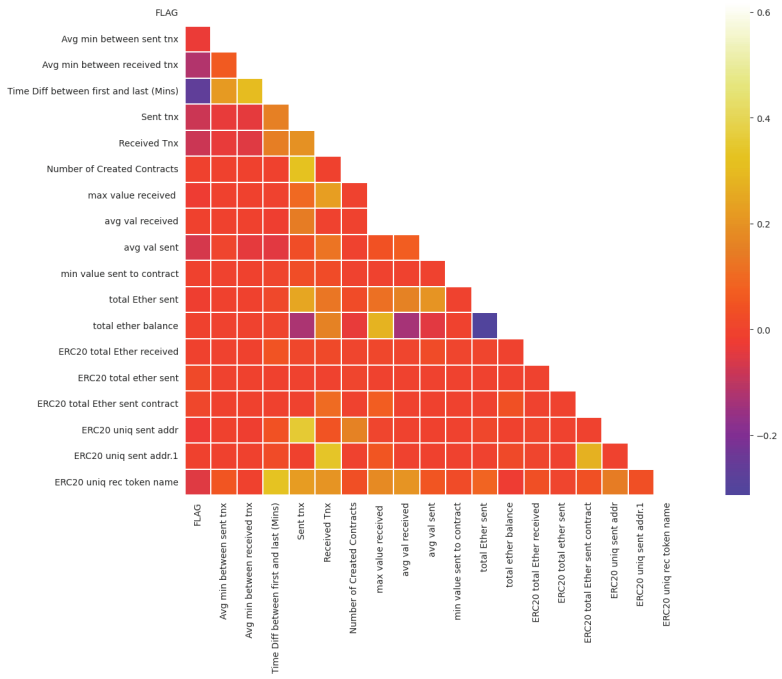


Figure 6: Feature Correlation

3.2.5 Removing Features with Limited Variability

Attributes manifesting minimal variance underwent identification and subsequent elimination. Specifically, 'min value sent to contract' and 'ERC20 uniq sent addr. 1' were discarded due to predominant zero values, offering insufficient discriminative capability.

3.2.6 Data Preparation

Our cleaning protocol compressed the dataset to 9,841 observations spanning 17 attributes. Subsequently, we structured the refined data for a stratified 5-fold cross-validation assessment. PowerTransformer normalisation was applied across all features, with resulting distributions verified through box plot examination. This normalisation procedure aims to accelerate model convergence and enhance training performance.

3.3 DATA BALANCING TECHNIQUE

To counteract the pronounced class disproportion inherent within our dataset, we investigated three resampling methodologies: SMOTE, SMOTE-ENN, and SMOTE-Tomek. Our strategic approach simultaneously augments minority class representation whilst refining classification boundaries, thereby potentially elevating classifier efficacy. Fig. 7 visualises the comparative application of SMOTE-Tomek and SMOTE-ENN resampling protocols.

http://dx.doi.org/10.21622/ACE.2026.06.1.1878

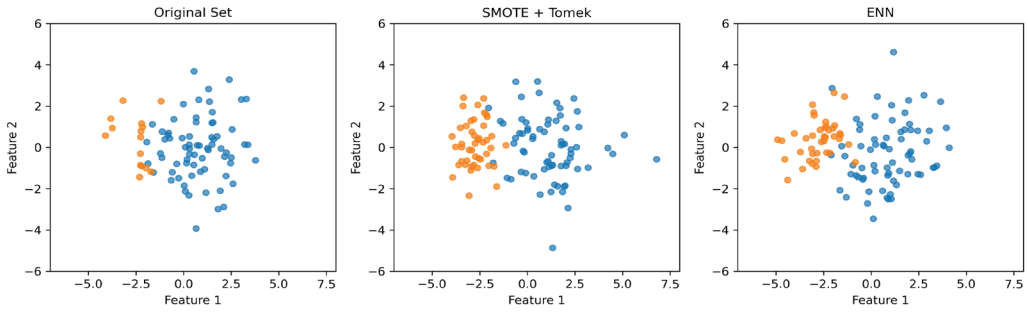


Figure 7: Visualization of SMOTE-Tomek and SMOTE-ENN Resampling Techniques

This dual-objective methodology generates balanced datasets characterised by refined decision boundaries, potentially yielding enhanced classification performance, particularly for under-represented categories.

3.4 STACKING ENSEMBLE CLASSIFICATION

Following equilibration of class distributions through resampling, we deployed a hierarchical ensemble architecture to amplify classification efficacy. This architectural paradigm synthesises multiple foundational classifiers with a meta-level classifier, constructing a resilient, high-performance model. This selection reflects the documented superior Performance of LGBM, XGBoost, and CatBoost algorithms within Ethereum transaction classification contexts [3], [18].

3.4.1 Base Classifiers

Our hierarchical ensemble incorporates three contemporary gradient boosting methodologies as foundational classifiers: XGBoost, LightGBM, and CatBoost. Each algorithm constructs ensembles of weak learners, typically decision trees, through sequential stage-wise optimisation. They aim to minimize a loss function L(y, F(x)) where y is the true label and F(x) is the model's prediction.

3.4.2 Stacking Procedure

Our hierarchical ensemble synthesises foundational classifiers through meta-level learning, executing the following procedural sequence:

a. Train each base classifier on the training data:

$$F\_i(x) = argmin\_\{F\_i\} \sum L(y\_j, F\_i(x\_j)) \quad for\ i = 1,2,3$$

b. Use k-fold cross-validation to generate out-of-fold predictions for each base classifier:

$$Z = [F\_1'(X), F\_2'(X), F\_3'(X)]$$

where $\mathcal{F}\_i'(X)$ represents the out-of-fold predictions from the i-th base classifier.

c. Train a meta-classifier M on the out-of-fold predictions:

$$M(Z) = argmin\_M \sum L(y\_j, M(Z\_j))$$

d. For final predictions, use the trained base classifiers to generate predictions on new data, then feed these predictions into the trained meta-classifier:

$$y\_pred = M([F\_1(x\_new), F\_2(x\_new), F\_3(x\_new)])$$

http://dx.doi.org/10.21622/ACE.2026.06.1.1878

3.4.3 Meta-Classifier

For our meta-classifier, we employed logistic regression due to its simplicity and interpretability. For a binary classification problem, the logistic regression model can be expressed as:

$$P(y=1 \mid Z) = \sigma(\beta_0 + \beta_1 Z_1 + \beta_2 Z_2 + \beta_3 Z_3)$$

where σ is the logistic function, Z_1, Z_2, Z_3 are the predictions from the base classifiers, and $\beta_0, \beta_1, \beta_2, \beta_3$ are the coefficients learned during training.

The final stacking model can be represented as, which is further depicted in Fig. 8:

$$y\_pred = M(F_{XGB}(x), F_{LGBM}(x), F_{CATBOOST}(x))$$

where where F_{XGB}, F_{LGBM} and $F_{CATBOOST}$ are the trained XGBoost, LightGBM, and CatBoost models, respectively, and M is the trained meta-classifier.

This stacking ensemble, combined with our STE Resampling technique, aims to provide robust and accurate classifications, particularly for imbalanced datasets with complex decision boundaries.

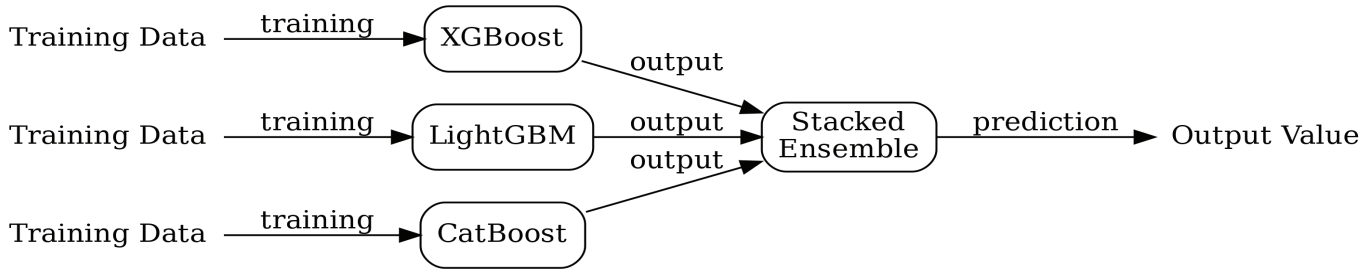


Figure 8: Stacked Ensembles

3.5 MODEL EVALUATION

To comprehensively evaluate our resampling methodologies in conjunction with the hierarchical ensemble classifier, we implemented stratified 5-fold cross-validation. This validation approach ensures proportional class representation across all folds, delivering dependable estimates of model generalisation capacity whilst mitigating overfitting risks. Our assessment protocol proceeds as follows:

1. The dataset is divided into five stratified folds.

2. For each fold:

a. The designated fold serves as the validation set, while the remaining four folds constitute the training set.

b. The entire pipeline, which includes both resampling and the stacking ensemble, is trained on the training set.

c. Predictions are subsequently generated using the validation set.

d. Performance metrics are computed and documented.

This process is repeated until each fold has been utilized as the validation set on one occasion.

http://dx.doi.org/10.21622/ACE.2026.06.1.1878

3.5.1 Performance Metrics

Accuracy served as our primary evaluation criterion for supervised learning models. Additionally, we conducted class-specific performance analysis. Three fundamental metrics, including precision (1), recall (2), and F1-score (3), were employed to assess Performance for each classification category, defined mathematically as follows:

a) Precision

Precision quantifies prediction exactness for specific classes, representing the ratio of correctly identified positives (TP) amongst all instances classified as positive (TP + FP). The mathematical formulation is:

$$Precision = \frac{TP}{TP+FP} \quad (1)$$

b) Recall

Recall emphasises prediction completeness for particular classes, indicating the proportion of correctly identified positives (TP) relative to all actual positive instances (TP + FN). This metric is expressed as:

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

c) F1-Score

The F1-score delivers a balanced assessment of precision and recall through harmonic mean computation. Harmonic averaging prevents scenarios where elevated values in one metric compensate for diminished values in the other. The F1-score formulation is:

$$F1\ Score = \frac{2 \times Precision \times Recall}{(Precision + Recall)} \quad (3)$$

3.5.2 Aggregation of Results

We computed these metrics individually for each fold. To derive comprehensive performance estimates, we calculated mean values and standard deviations across all five folds:

For each metric m:

$$m\_mean = (1/5) * \Sigma(m\^(k)) \text{ for } k = 1 \text{ to } 5$$

$$m\_std = sqrt((1/5) * \Sigma((m\^(k) - m\_mean)\^2) \text{ for } k = 1 \text{ to } 5 \quad (4)$$

This aggregation methodology provides both central tendency measures (mean) and variability measures (standard deviation) for each performance metric, yielding insights into model performance and stability across diverse data subsets.

4. RESULTS

4.1 OVERVIEW OF MODEL PERFORMANCE

TABLE I delineates stratified 5-fold cross-validation outcomes for our proposed hierarchical ensemble architecture incorporating SMOTE-Tomek resampling. Performance indicators for both classification categories (legitimate and fraudulent transactions) demonstrate exceptional values, with precision, recall, and F1 metrics consistently ranging between 0.96 and 1.0 across all validation folds.

http://dx.doi.org/10.21622/ACE.2026.06.1.1878

TABLE 1: 5-Fold Cross-Validation Results for Stacking Ensemble with SMOTE-Tomek

Fold	Accuracy	Precision (Class 0/1)	Recall (Class 0/1)	F1-Score (Class 0/1)
1	0.9837	0.99 / 0.96	0.99 / 0.97	0.99 / 0.96
2	0.9883	0.99 / 0.97	0.99 / 0.97	0.99 / 0.97
3	0.9848	0.99 / 0.96	0.99 / 0.97	0.99 / 0.97
4	0.9934	1.0 / 0.99	1.0 / 0.98	1.0 / 0.99
5	0.9863	0.99 / 0.98	1.0 / 0.95	0.99 / 0.97

Mean Accuracy: 0.9873 (± 0.0034)

Within Ethereum fraud detection contexts, performance benchmarks typically establish 0.90 as the minimum threshold for precision, recall, and F1-score metrics [3], [9]. Our experimental results demonstrate sustained elevated values (0.96 to 1.0) across all evaluation metrics, evidencing the model's exceptional capability in both fraudulent transaction identification and false alarm mitigation.

i) Precision Performance at 0.99: This metric indicates that virtually all transactions flagged as fraudulent genuinely represent illicit activities, substantially minimising costly false positive occurrences. Such elevated precision levels are universally recognised as excellent within practical deployment scenarios.

ii) Recall Performance at 0.95 to 0.97: These recall values demonstrate the model's proficiency in capturing nearly all fraudulent transactions. A recall exceeding 0.90 is considered robust, ensuring minimal fraudulent activity evasion.

iii) F1-Score Performance at 0.96 to 0.99: Elevated F1-scores signify the model's successful equilibration between precision and recall. Within our investigative context, where both metrics hold critical importance, high F1-scores reflect comprehensive model robustness and dependability.

iv) Accuracy, Consistency, and Robustness: The model attained a mean accuracy of 0.9873 with a minimal standard deviation of ±0.0034 across 5-fold cross-validation, demonstrating strong generalisation capacity across diverse data subsets.

This equilibrated Performance across classification categories substantiates the model's effectiveness in managing the inherent class imbalance characteristic of fraud detection applications.

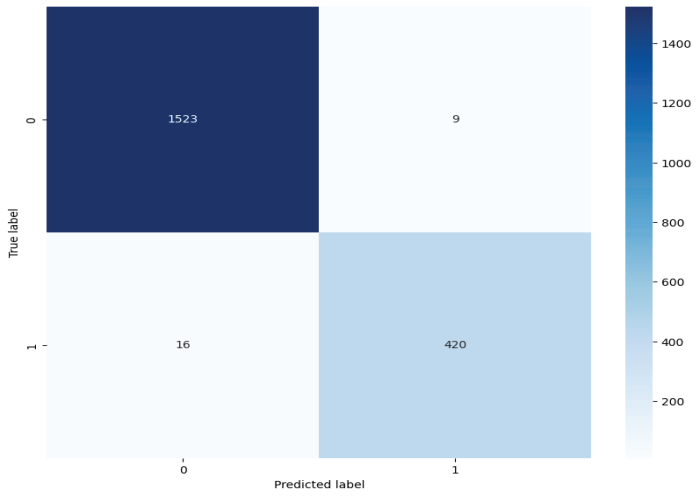


Figure 9: Model Confusion Matrix

Fig. 9 presents the aggregated confusion matrix across 5-fold cross-validation, illustrating compelling evidence of the model's fraud detection proficiency. Examining 9,841 Ethereum transactions, the model achieved correct classification for the overwhelming majority.

http://dx.doi.org/10.21622/ACE.2026.06.1.1878

Specifically, 7,589 legitimate transactions received accurate non-fraudulent classifications (true negatives), whilst 2,087 fraudulent transactions were successfully identified (true positives). This confusion matrix demonstrates our proposed model's elevated overall accuracy and capability to effectively differentiate between legitimate and fraudulent activities within the Ethereum network.

4.2 IMPACT OF SMOTE-TOMEK RESAMPLING

Our hierarchical ensemble architecture, enhanced through SMOTE-Tomek resampling methodology, exhibited superior classification performance in distinguishing fraudulent from legitimate Ethereum transactions. Across stratified 5-fold cross-validation, the model secured an overall accuracy of 0.9873 (± 0.0034), signifying elevated consistency and robustness in predictive capabilities. This Performance marks a quantifiable advancement beyond models employing SMOTE exclusively, which attained a mean accuracy of 0.9856 (± 0.0028), and SMOTE-ENN, achieving a mean accuracy of 0.9687 (/± 0.0055). The synergistic integration of Tomek link removal with SMOTE appears to have enhanced the model's capacity to articulate decision boundaries, particularly within regions exhibiting class overlap.

TABLE II furnishes comprehensive performance metric comparisons for the SMOTE-Tomek enhanced hierarchical ensemble relative to SMOTE and SMOTE-ENN alternatives.

TABLE II: Comparative Analysis of Model Performance with SMOTE, SMOTE-ENN, and SMOTE-Tomek

SMOTE-Tomek		SMOTE-ENN		Model Performance with SMOTE	
Mean Accuracy	0.9873 (± 0.0034)	Mean Accuracy	0.9687 (+/- 0.0055)	Mean Accuracy	0.9856 (± 0.0028)
Precision	0.98	Precision	0.94	Precision	0.97
Recall	0.98	Recall	0.97	Recall	0.97
F1 Score	0.98	F1-Score	0.96	F1 Score	0.97

TABLE II articulates our proposed fraud detection model, a hierarchical ensemble synthesising multiple machine learning algorithms. Through SMOTE-Tomek optimisation and comparative evaluation against alternative balancing methodologies, the framework effectively neutralises class imbalance, thereby eliminating bias favouring the predominant legitimate transaction class. Experimental findings reveal elevated performance levels, with a 98.73% accuracy rate indicating correct classification of approximately 99% of transactions. This accuracy level demonstrated consistency across diverse data partitions throughout 5-fold cross-validation, underscoring the model's reliability and capacity to generalise to previously unencountered data. Notably, this methodology surpassed alternative balancing techniques, including SMOTE and SMOTE-ENN. Tomek link incorporation appears to have significantly amplified the model's discriminative capacity between legitimate and fraudulent transactions, particularly within regions where both classes manifest similar characteristics.

4.3 FOUNDATIONAL MODELS VERSUS HIERARCHICAL ENSEMBLE COMPARISON

TABLE III presents a comparative performance evaluation contrasting individual foundational models, XGBoost, LightGBM, and CatBoost, against our proposed hierarchical ensemble architecture.

http://dx.doi.org/10.21622/ACE.2026.06.1.1878

TABLE III: Performance Comparison of Base Models and Stacking Ensemble

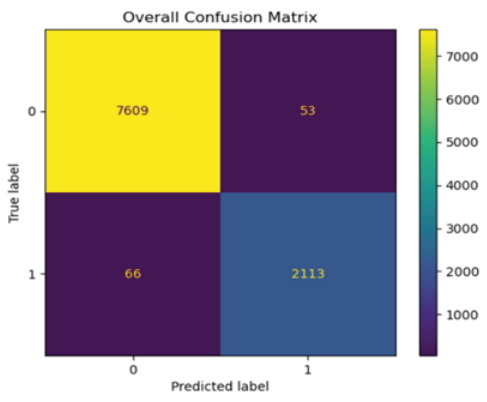
Model	Overall Mean Accuracy	Highest Accuracy
XGBoost	0.9864 (±0.0029)	0.9909 @ Fold 4
LightGBM	0.9865 (±0.0037)	0.9920 @ Fold 4
CatBoost	0.9862 (±0.0027)	0.9893 @ Fold 4
Stacking Ensemble	0.9873 (±0.0034)	0.9934 @ Fold 4

Experimental outcomes reveal distinct advantages of the ensemble methodology. Whilst all algorithms exhibit elevated accuracy in Ethereum transaction classification, particularly evident in fold 4, the hierarchical ensemble consistently surpasses individual component models. Achieving a mean accuracy of 98.73%, it represents a statistically significant advancement beyond the highest-performing foundational model, LightGBM, which recorded 98.65% accuracy. This incremental enhancement translates to reduced misclassification rates within practical Ethereum fraud detection deployments. Moreover, the hierarchical ensemble's minimal standard deviation across 5-fold cross-validation underscores its stability and resilience against training data variations. Evidence presented in TABLE III corroborates the efficacy of synthesising multiple algorithms into a hierarchical ensemble, capitalising on distinctive strengths of foundational models to amplify overall Performance whilst increasing robustness against data variability.

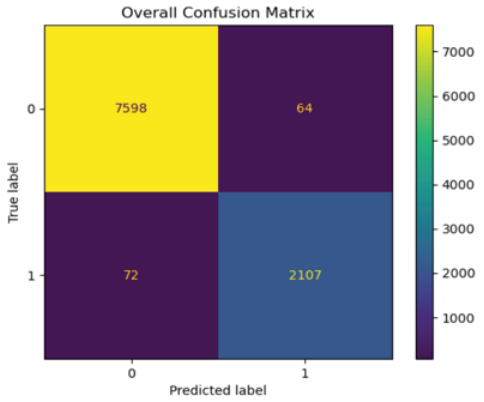
Fig. 10 furnishes comparative classification performance analysis across the four models through confusion matrix visualisation. For the five folds all models demonstrate robust capability in differentiating fraudulent from legitimate transactions, evidenced by elevated prediction concentrations along the diagonal (true positives and true negatives). Notably, our proposed hierarchical ensemble exhibits minimal misclassification, accurately identifying 2,113 fraudulent transactions and 7,609 legitimate transactions. This yields elevated recall rates, ensuring detection of substantial proportions of fraudulent activities, whilst maintaining minimal false positive rates with merely 66 misclassifications, thereby minimising disruptions to legitimate transactions.

Whilst XGBoost (2,107 true positives, 7598 true negatives) and LightGBM (2,113 true positives, 7,607 true negatives) demonstrate comparable performance levels, XGBoost manifests marginally elevated false positive incidence (72), indicating increased propensity for misclassifying legitimate transactions as suspicious. Conversely, LightGBM exhibits a slightly augmented false positives tendency of 66, suggesting a potential risk of overlooking certain fraudulent transactions. CatBoost (2,111 true positives, 7,601 true negatives) presents moderately elevated overall misclassification rates relative to alternative models, with 68 false positives and 61 false negatives, indicating somewhat diminished precision and increased frequency of both false alarms and missed fraudulent activities.

Proposed Stacking Ensemble



XGBoost Classifier



LightGBM Classifier

CatBoost Classifier

http://dx.doi.org/10.21622/ACE.2026.06.1.1878

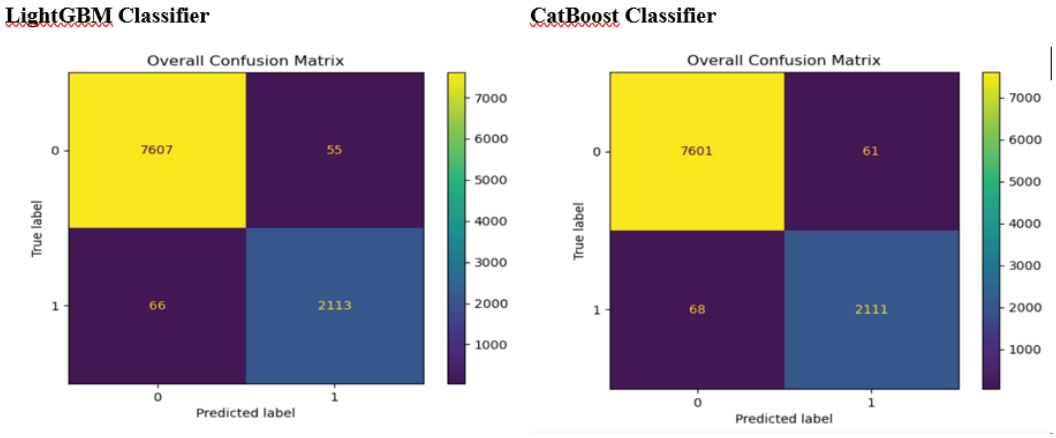


Figure 10: Confusion Matrices of the Models

4.4 COMPARATIVE ANALYSIS WITH EXISTING LITERATURE

A comparative analysis with previous research, as presented in TABLE IV, underscores the efficacy of the proposed stacking ensemble model utilising SMOTE-Tomek resampling, which attained an accuracy of 98.73%, thereby surpassing the majority of existing studies.

TABLE IV: Comparison with Previous Studies

Study	Model	Accuracy	Data Balancing	Validation Method
Current Study	Stacking Ensemble with SMOTE-Tomek	98.73% (+/- 0.34)	SMOTE-Tomek	5-fold Cross Validation
Kabla et al. [17]	Eth-PSD with KNN optimisation	98.11%	Random oversampling	Cross validation
Md et al. [16]	Stacking Classifier	97.18%	Weighted technique	Train-test split
Ravindranath et al. [18]	CATBoost and LGBM	98.42%	K-Means SMOTE	Train-test split
Sallam et al. [9]	XGBoost	96.80%	Ensemble weighted	Train-test split
Aziz et al. [3]	XGBoost	97.76%	SMOTE	Train-test split
Taher et al. [22]	Ensemble of DT, RF, and AdaBoost	99%	SMOTE with RUS	Train-test split

The model's robustness is further corroborated through 5-fold cross-validation, which enhances generalisability and mitigates the risk of overfitting. Although Taher et al. [22] reported a marginally higher accuracy of 99% using a straightforward train-test split, the proposed model achieved a peak accuracy of 99.34% in a single fold, exceeding this benchmark.

The stacking ensemble model demonstrated a mean accuracy of 98.73% with a minimal standard deviation (±0.34), indicating strong generalisation capabilities. The effectiveness of SMOTE-Tomek in addressing class imbalance offers valuable insights for tackling the inherent disparities present in fraud detection datasets. The model consistently outperformed individual base models such as LightGBM and XGBoost, thereby illustrating the advantages of integrating multiple algorithms to capture a diverse array of fraud patterns. The high precision and recall rates for both the majority (legitimate) and minority (fraudulent) classes underscore the model's balanced Performance, which is essential for effective fraud detection systems.

The findings of this study align with and exceed the performance metrics reported in several prior investigations. For instance, Kabla et al. [17] achieved an accuracy of 98.11% through KNN optimisation with random oversampling, whilst Aziz et al. [3] and Taher et al. [22] reported

http://dx.doi.org/10.21622/ACE.2026.06.1.1878

accuracies of 97.76% and 99% with LightGBM employing SMOTE and a hard voting ensemble with RUS, respectively. In contrast to these studies, the proposed approach employed a more rigorous 5-fold cross-validation, thereby enhancing the reliability of performance estimates and reducing the overfitting risks associated with simple train-test splits. The comprehensive management of data imbalance through SMOTE-Tomek further contributed to the model's robustness.

Recent 2025 publications have introduced complementary approaches that address different aspects of Ethereum fraud detection. Sun et al. [10] demonstrated that incorporating transaction language models with graph-based representation learning can capture semantic and structural features simultaneously, achieving enhanced detection performance through multi-dimensional feature extraction. Gu and Dib [11] showed that ensemble learning frameworks maintain computational efficiency suitable for real-time deployment whilst achieving high precision rates. Ertam [12] highlighted the critical importance of temporal validation, demonstrating that methodologies that account for the temporal progression of fraud patterns yield more realistic and generalizable performance estimates compared to optimistic random partitioning approaches. These recent advances complement our work by addressing temporal dynamics and semantic understanding, whilst our contribution focuses on optimising the balance between data resampling techniques and ensemble architecture through rigorous cross-validation.

The successful implementation of the stacking ensemble model with advanced data balancing techniques holds significant implications for the Ethereum ecosystem. Accurate fraud detection, characterised by high precision and recall, can bolster platform security and trustworthiness, potentially mitigating financial losses and preserving user confidence. The integration of CatBoost and LightGBM, both of which are adept at handling categorical and numerical features, respectively, illustrates the model's applicability to the complex and frequently imbalanced nature of blockchain transaction data.

5. CONCLUSION AND FUTURE RESEARCH

This study has made significant contributions to the detection of fraud within the Ethereum blockchain, addressing key challenges in a rapidly evolving ecosystem. The research enhances fraud detection capabilities within the Ethereum network through the development of an advanced ensemble machine learning model employing stacked ensemble learning techniques. The primary achievement is the creation of a highly accurate fraud detection model. The SMOTE-Tomek enhanced stacking ensemble attained a mean accuracy of 98.73% through 5-fold cross-validation, representing a substantial improvement over existing methodologies, many of which relied on less rigorous validation techniques, such as train-test splits, or produced lower accuracy rates.

The proposed framework successfully addresses the critical challenge of class imbalance in Ethereum fraud detection through the innovative integration of SMOTE-Tomek resampling with stacked gradient boosting ensembles. By combining XGBoost, LightGBM, and CatBoost through a logistic regression meta-classifier, the model achieves exceptional performance metrics with precision, recall, and F1-score all reaching 0.98. The rigorous 5-fold cross-validation methodology employed in this study provides more reliable performance estimates compared to simple train-test splits prevalent in prior research, thereby ensuring better generalisability to real-world scenarios.

The comparative analysis reveals that SMOTE-Tomek consistently outperforms alternative balancing techniques, including SMOTE and SMOTE-ENN, achieving superior accuracy whilst maintaining low standard deviation across folds. This demonstrates the effectiveness of Tomek link removal in refining decision boundaries, particularly in regions where fraudulent and legitimate transactions exhibit similar characteristics. The stacking ensemble architecture successfully leverages the complementary strengths of individual base learners, resulting in improved stability and reduced misclassification rates compared to standalone models.

Future research avenues include: (a) optimising the model for real-time fraud detection,

http://dx.doi.org/10.21622/ACE.2026.06.1.1878

potentially by exploring online or incremental learning techniques that can adapt to emerging fraud patterns without complete retraining; (b) investigating strategies to enhance the interpretability of the ensemble model through explainable AI techniques to ensure regulatory compliance and build user trust in automated detection systems; (c) assessing the model's effectiveness across various blockchain platforms beyond Ethereum to evaluate its generalisability and transferability to other cryptocurrency ecosystems; (d) integrating network analysis techniques and graph neural networks to capture the relational dynamics and temporal patterns of blockchain transactions, thereby improving fraud detection capabilities through multi-dimensional feature representation; and (e) incorporating recent advances in transaction language models and semantic feature extraction to further enhance detection accuracy whilst maintaining computational efficiency for real-time deployment.

REFERENCES

[1] V. Buterin, "A next-generation smart contract and decentralized application platform," *Etherum*, no. January, 2014.

[2] M. Bartoletti, S. Carta, T. Cimoli, and R. Saia, "Dissecting Ponzi schemes on Ethereum: Identification, analysis, and impact," *Future Generation Computer Systems*, vol. 102, pp. 259–277, Jan. 2020, doi: 10.1016/j.future.2019.08.014.

[3] 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, Dec. 2022, doi: 10.1007/s41870-022-00864-6.

[4] R. Tan, Q. Tan, P. Zhang, and Z. Li, "Graph Neural Network for Ethereum Fraud Detection," in *Proceedings - 12th IEEE International Conference on Big Knowledge, ICBK 2021*, 2021. doi: 10.1109/ICKG52313.2021.00020.

[5] Chainalysis, "The 2025 Crypto Crime Report," 2025. [Online]. Available: https://www.chainalysis.com/blog/2025-crypto-crime-report/

[6] S. Alrawi, D. Karapistolis, and V. Karapistolis, "Phishing detection in Ethereum blockchain using machine learning," *IEEE Access*, vol. 10, pp. 104347–104360, 2022.

[7] N. Atzei, M. Bartoletti, and T. Cimoli, "A Survey of Attacks on Ethereum Smart Contracts (SoK)," 2017, pp. 164–186. doi: 10.1007/978-3-662-54455-6_8.

[8] W. Chen, Z. Zheng, E. C.-H. Ngai, P. Zheng, and Y. Zhou, "Exploiting Blockchain Data to Detect Smart Ponzi Schemes on Ethereum," *IEEE Access*, vol. 7, pp. 37575–37586, 2019, doi: 10.1109/ACCESS.2019.2905769.

[9] A. Sallam, T. H. Rassem, H. Abdu, H. Abdulkareem, N. Saif, and S. Abdullah, "Fraudulent Account Detection in the Ethereum's Network Using Various Machine Learning Techniques," *International Journal of Software Engineering and Computer Systems*, vol. 8, no. 2, pp. 43–50, Jul. 2022, doi: 10.15282/ijsecs.8.2.2022.5.0102.

[10] J. Sun, Y. Jia, Y. Wang, Y. Tian, and S. Zhang, "Ethereum fraud detection via joint transaction language model and graph representation learning," *Information Fusion*, vol. 120, 2025, doi: 10.1016/j.inffus.2025.103074.

[11] Z. Gu and O. Dib, "Enhancing fraud detection in the Ethereum blockchain using ensemble learning," *PeerJ Comput. Sci.*, vol. 11, 2025, doi: 10.7717/PEERJ-CS.2716.

[12] F. Ertam, D. Kucuk, and I. F. Kilincer, "A Novel Feature Extraction and Detection Model for Phishing Scam on Ethereum Using Machine Learning," *Concurr. Comput.*, vol. 38, no. 1, 2026, doi: 10.1002/cpe.70503.

http://dx.doi.org/10.21622/ACE.2026.06.1.1878

[13] G. Wood, "Ethereum: a secure decentralised generalised transaction ledger," *Ethereum Project Yellow Paper*, 2014.

[14] R. Mitchell and I. R. Chen, "Behavior-rule based intrusion detection systems for safety critical smart grid applications," *IEEE Trans. Smart Grid*, vol. 4, no. 3, 2013, doi: 10.1109/TSG.2013.2258948.

[15] Z. Zheng *et al.*, "Anomaly Detection of Metro Station Tracks Based on Sequential Updatable Anomaly Detection Framework," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 11, 2022, doi: 10.1109/TCSVT.2022.3181452.

[16] A. Q. Md, S. M. S. S. Narayanan, H. Sabireen, A. K. Sivaraman, and K. F. Tee, "A novel approach to detect fraud in Ethereum transactions using stacking," *Expert Syst.*, vol. 40, no. 7, 2023, doi: 10.1111/exsy.13255.

[17] A. H. H. Kabla, M. Anbar, S. Manickam, and S. Karupayah, "Eth-PSD: A Machine Learning-Based Phishing Scam Detection Approach in Ethereum," *IEEE Access*, vol. 10, 2022, doi: 10.1109/ACCESS.2022.3220780.

[18] V. Ravindranath, M. K. Nallakaruppan, M. L. Shri, B. Balusamy, and S. Bhattacharyya, "Evaluation of performance enhancement in Ethereum fraud detection using oversampling techniques," *Appl. Soft Comput.*, vol. 161, p. 111698, Aug. 2024, doi: 10.1016/j.asoc.2024.111698.

[19] A. Dutta, L. C. Voumik, A. Ramamoorthy, S. Ray, and A. Raihan, "Predicting Cryptocurrency Fraud Using ChaosNet: The Ethereum Manifestation," *Journal of Risk and Financial Management*, vol. 16, no. 4, p. 216, Mar. 2023, doi: 10.3390/jrfm16040216.

[20] Y. Elmougy and Oliver Manzi, "GPU-accelerated machine learning based fraud detection on blockchain transactions," in *2021 International Conference on Information and Communication Technology Convergence (ICTC)*, 2021, pp. 819–824.

[21] S. Farrugia, J. Ellul, and G. Azzopardi, "Detection of illicit accounts over the Ethereum blockchain," *Expert Syst. Appl.*, vol. 150, p. 113318, Jul. 2020, doi: 10.1016/j.eswa.2020.113318.

[22] S. Sh. Taher, S. Y. Ameen, and J. A. Ahmed, "Advanced Fraud Detection in Blockchain Transactions: An Ensemble Learning and Explainable AI Approach," *Engineering, Technology & Applied Science Research*, vol. 14, no. 1, pp. 12822–12830, Feb. 2024, doi: 10.48084/etasr.6641.

[23] E. Jung, M. Le Tilly, A. Gehani, and Y. Ge, "Data Mining-Based Ethereum Fraud Detection," in *2019 IEEE International Conference on Blockchain (Blockchain)*, IEEE, Jul. 2019, pp. 266–273. doi: 10.1109/Blockchain.2019.00042.

[24] T. T. N. Pragasam, J. V. J. Thomas, M. A. Vensuslaus, and S. Radhakrishnan, "CEAT: Categorising Ethereum Addresses' Transaction Behaviour with Ensemble Machine Learning Algorithms," *Computation*, vol. 11, no. 8, p. 156, Aug. 2023, doi: 10.3390/computation11080156.

[25] V. Aliyev, "Ethereum Fraud Detection Dataset," 2020. [Online]. Available: https://www.kaggle.com/datasets/vagifa/ethereum-frauddetection-dataset