Journal of Computer Science

Research Article

Optimized XGBoost for Ethereum Fraud Detection: A Cost-Sensitive Approach

Supriya P., Rubah Sheriff, Shreya Padaki, Suchi V. Yadav and Varsha Thaku

Department of Artificial Intelligence and Machine Learning, B.M.S. College of Engineering, Bengaluru, Karnataka, India

Article history
Received: 22-04-2025
Revised: 25-06-2025
Accepted: 10-07-2025

Corresponding Author:
Supriya P.
Department of Artificial Intelligence and Machine Learning, B.M.S. College of Engineering, Bengaluru, Karnataka, India
Email: supriyap.mel@bmsce.ac.in

Abstract: In today's technologically advancing world, many fields from finance to healthcare and education are shifting toward a digital and decentralized format. A significant transformation is underway with the currency of the masses. Blockchain-based cryptocurrencies like Bitcoin and Ethereum allow users to generate fungible tokens anonymously through smart contracts. However, these features also facilitate illicit transactions and cybercrimes like fraud, phishing, and money laundering. The proposed work explores the identification of suspicious transactions on the Ethereum blockchain by leveraging advanced machine-learning techniques. An Extreme Gradient Boosting (XGBoost) classifier is optimized for spotting unauthorized or malicious transactions, exploring features like transaction patterns and value anomalies. Feature scaling and log transformations normalize skewed distributions, while rigorous model training and hyperparameter tuning enhance the system's precision, recall, and overall accuracy. Other aids, such as feature importance rankings, precision-recall curves, and diagnostic statistics, provide useful information on fraud patterns. Evaluation of the model shows that integrating cost-sensitive learning significantly reduces false positives, from 51 to 44, representing a 13.7% decrease, which enhances practical usability by minimizing false alerts and manual verification efforts. Although there was a slight increase in false negatives (from 14 to 15), the overall classification accuracy improved. The model demonstrated strong performance in managing class imbalance which is common in fraud detection contexts.

Keywords: Ethereum, Fraud Detection, Blockchain, XGBoost, Classification, Anomaly Detection, Feature Engineering, Hyperparameter Tuning, Evaluation Metrics

Introduction

The large-scale implementation of blockchain technology has resulted in the exponential growth of cryptocurrency transactions, especially in Ethereum. With the growing adoption of blockchain technology, the volume of cryptocurrency transactions on the Ethereum network has significantly surged. Nevertheless, because of its pseudonymous nature, blockchain technology has become a popular target for illicit activities such as money laundering, phishing attacks, and Ponzi schemes (Tripathy *et al*., 2024). Identifying illicit accounts remains a significant challenge due to the vast volume of cryptocurrency transactions and the absence of clear, identifiable user information (Ding *et al*., 2024). The rarity of fraudulent transactions, combined with the high dimensionality of transaction features and the volatility of cryptocurrency markets, poses significant challenges for accurate and reliable fraud detection models (Jiang, *et al*., 2024).

Scholars have turned to Machine Learning (ML) (Kerr *et al*., 2023), graph-based techniques and neural networks which mimic the biological neural systems (Dutta *et al*., 2023) for effective solutions because traditional fraud detection methods are unable to handle the dynamic nature of blockchain fraud, these suggested techniques provide insights for investors, regulators about currency types, fraud cases and risk factors.

Machine learning algorithms have shown significant promise in detecting illegal activities by analysis of transactional behavior, frequency, and patterns (Y

Elmougy, and Liu, 2023). The ML component leverages algorithms to classify transactions, while the blockchain provides tamper-proof and transparent verification (Sultana *et al*., 2023). Machine Learning plays a transformative role in cryptocurrency markets by revolutionizing trading strategies, enhancing risk management, improving fraud detection, and enabling advanced market analytics, offering valuable insights for businesses seeking to adopt these technologies effectively (Islam *et al.*, 2025). Imbalanced classification methods and explainable AI models have been investigated to enhance fraud detection accuracy while maintaining the ability to explain. The paper discusses the application of a new fraud detection system for Ethereum transactions. Sophisticated machine learning algorithms with extensive data pre-processing and feature selection methods reveal subtle patterns in the anonymized blockchain transactions (Elmougy and Liu, 2023). These, combined with adaptive learning features that identify changing patterns of fraudulent transactions, increase the security and integrity of the Ethereum network. The goal is to create a strong framework that can identify multiple forms of fraudulent activity, such as phishing attacks and token thefts, and constantly adapt to new cybercrime attack schemes (Taher *et al*., 2024), creating a more secure ecosystem for users (Venkatesh *et al*., 2024).

The rapid adoption of cryptocurrencies has fueled financial technology innovations as well as posed fraud detection issues. Cryptocurrency transactions are susceptible to cybercrimes like phishing, Ponzi schemes, and money laundering. Researchers have responded by developing advanced fraud detection systems through Machine Learning and Deep Learning paradigms. Different research pieces have examined the identification and mitigation of fraudulent transactions in cryptocurrency systems, particularly on Ethereum.

A study was conducted to evaluate various machine learning techniques for classifying fraudulent cryptocurrency transactions (Tripathy *et al.*, 2024). They demonstrated that XGBoost achieved the highest performance, with an accuracy of 98%, outperforming several other models such as Random Forest, Logistic Regression, K-Nearest Neighbors, AdaBoost, and Support Vector Machines (SVM). The study emphasizes that classification-based machine learning approaches can effectively identify suspicious activities by leveraging behavior-derived transaction features. Another study presents a Directed Multigraph-Based Approach (DIAM) for discovering illegitimate accounts within crypto networks (Ding *et al*., 2024). Using Gated Recurrent Units (GRU) and Multilayer Perceptron (MLP), their algorithm excelled against the rest for classifying the existence of illicit accounts, presenting an example of what graph learning to counteract fraud can look like.

Another major obstacle to cryptocurrency fraud detection is anonymizing transactions. Jiang *et al*. (2024) addressed this issue by using the Synthetic Minority Oversampling Technique (SMOTE) for data imbalance and Bayesian optimization for improving fraud classification accuracy. Their work proved that Machine Learning models were able to identify large patterns of relevance to fraud using the anonymity of the cryptocurrency transactions. Also, ensemble learning approaches to detect fraudulent Bitcoin transactions proved that assembling multiple classifiers facilitates anomaly detection, specifically identification of a cluster of malicious transactions (Elmougy and Liu, 2023).

The adaptive nature of fraud makes it challenging for adaptive learning mechanisms, so that fraud detection is successful. Taher *et al*. (2024) employed ensemble learning combined with explainable AI to develop a strong fraud detection system that could learn to adapt to changing fraudulent patterns. They did mention, however, that the high computational cost is still a limitation for real-time fraud detection. Likewise, ChaosNet, a new technique applied chaos theory in Machine Learning models for recognizing fraud in Ethereum transactions (Dutta *et al*., 2023). Their model was highly resistant to adversarial attacks and efficient in identifying fraudulent behavior for anomalous transaction patterns.

Pattern identification is the most important factor in identifying fraudulent transactions in blockchain networks. Sultana *et al*. (2023) combined Machine Learning techniques with blockchain analysis, achieving more than 99% accuracy in identifying fraudulent transactions. Their work emphasized the necessity of addressing imbalanced data through SMOTE and feature selection techniques. Walavalkar *et al*. (2024) suggested a token-based approach for Ethereum fraud detection using sophisticated data pre-processing and classification methods. Their approach was centered on detecting suspicious token transactions, which worked in fraud detection.

Effective real-time fraud detection is critical due to the heavy flow of cryptocurrency transactions. Kerr *et al*. (2023), analyzed several cases of financial fraud in crypto markets and offered a risk assessment framework for blockchain transactions. They highlighted the use of hybrid strategies that incorporate standard financial risk measures with Machine learning-based anomaly detection models. Islam *et al*. (2025), examined how machine learning algorithms impact cryptocurrency fraud detection and identified key methods that organizations can utilize to improve security.

Studies have also been aimed at identifying certain forms of fraud that exist within the Ethereum environment. Zhang *et al*. (2024) proposed FRAD, a ternary classification framework for the detection of

front-running attacks in Ethereum. FRAD classifies transactions related to front-running attacks correctly, which allows developers to counter each attack form with proper strategies. Additionally, Palaiokrassas *et al*. (2024), created a framework utilizing Machine Learning for detecting multichain Decentralized Finance (DeFi) fraud. Their method draws features from various chains, such as Ethereum, and uses models such as XGBoost and neural networks to detect malicious accounts engaging with DeFi protocols.

Notable contributions to detect fraud in Ethereum transactions include a Long Short-Term memory(LSTM) classifier to generate recommendations based on Classification Scores (Sureshbhai *et al*., 2020), One-Class Graph Neural Network based anomaly detection framework (Ibrahim *et al*., 2021), Heterogenous Graph Neural Networks (Kanezashi *et al*., 2022), the light gradient boosting machine(LGBM) algorithms (Aziz *et al*., 2022), XGBoost and Random Forest(RF) algorithms used for transaction classification(Ashfaq *et al*., 2022; Farrugia *et al*., 2020).

Review papers on blockchain technology and its transactions explained the vulnerability of networks to anomalies and fraud, the need to identify emerging trends to mitigate these challenges (Osterrieder *et al*., 2023; Krishna and Praveenchandar, 2022), while another study focused on the usage of GPU-accelerated machine learning models for designing automated fraud detection systems (Elmougy and Manzi, 2021). Hu *et al*. (2023) proposed a temporal weighted heterogeneous multigraph embedding approach to identify phishing scams. Hu *et al*. (2022) introduced SCSGuard, a deep learning framework for fraudulent smart contracts.

In summary, notable advancements have been made in the detection of cryptocurrency fraud, and Machine Learning Methods (Ibrahim *et al*., 2021; Sallam *et al*., 2022) and Deep Learning methods (Giantsidi and Tarantola, 2025) have played a significant part. The examined studies emphasize the significance of effective fraud detection, dealing with data imbalance, adaptive learning, and pattern identification in stopping illegal transactions. Despite these advances, computational efficiency and the dynamic nature of fraud pose challenges. Future studies must emphasize improving real-time detection mechanisms and combining blockchain analytics with Machine learning-based solutions to improve cryptocurrency security.

Despite progress, there remains a gap in developing holistic fraud detection systems that combine graph-based features, cost-sensitive classification, and unsupervised anomaly detection to effectively handle class imbalance, dynamic fraud patterns, and transaction interconnectivity.

This paper aims to address this gap by proposing a robust, adaptive fraud detection framework for Ethereum. Cost-sensitive XGBoost classification was used to treat class imbalance (by penalizing false negatives), which allowed the model to improve its ability to discover rare but important instances. Graph-based feature augmentation techniques (i.e. PageRank & betweenness centrality) were used to augment relational/network level behavioral information in order for the model to exploit the structural relationships between entities. In addition, anomaly scores derived from unsupervised learning methods (i.e. Isolation Forest) were used to help identify previously unseen/evolutionary patterns that may not be adequately represented in labeled training set data. Finally, to ensure that model results were sufficiently robust and unbiased, Bayesian hyperparameter tuning in combination with nested cross-validation was used to provide the best possible hyperparameters while minimizing overfitting and/or evaluation bias.

The goal is to deliver a scalable and explainable system capable of detecting diverse types of fraudulent activity including phishing attacks, token theft, and smart contract abuse while adapting to emerging cybercrime tactics. This contributes to strengthening the security and trustworthiness of the Ethereum ecosystem.

Open Issues

Balancing Data in Fraud Detection Models

One of the largest challenges of fraudulent transaction detection in Ethereum is the enormous imbalance of datasets. Money laundering and Ponzi schemes are only a minority of all transactions, and it is difficult for machine learning models to learn useful patterns. Few studies (Jiang *et al*., 2024) explored an approach using the Synthetic Minority Over-sampling Technique (SMOTE) and cost-sensitive learning. Such approaches have a tendency to introduce artificial noise, leading to false positives and reducing the ability of the model to generalize real-world transactions. There are also consistently changing methods, forcing models to adapt dynamically. Creating advanced data-balancing techniques that preserve minority-class transaction integrity without compromising model accuracy is a key open problem in fraud detection.

Developing sophisticated data-balancing methods that protect the integrity of minority-class transactions without sacrificing model accuracy is an important open question in fraud detection.

Improving Real-Time Fraud Detection in Ethereum

With Ethereum processing thousands of transactions per second, real-time fraud detection is key to avoiding financial losses. Yet, the majority of current fraud detection models are based on computationally costly methods like deep learning and

ensemble learning, which cannot match the velocity of blockchain transactions. Conventional methods examine transaction patterns once they have been written to the blockchain, which is too late for fraud detection and poses greater financial risks. In addition, the implementation of fraud detection algorithms within Ethereum's transaction verification procedure is problematic in a decentralized platform like Ethereum. Fraud detection software has to walk a thin line between computational efficacy and precision, such that the suspicious transactions are flagged in real-time without delaying transactions appreciably.

Front-Running Attacks in Ethereum Identification and Prevention

Front-running is a serious security problem in Ethereum, whereby malicious traders use the transaction ordering mechanism to manipulate asset prices and achieve maximum personal benefits. Front-running attacks leverage Ethereum's public mempool, where transactions are stored temporarily before confirmation (Zhang *et al*., 2024). Attackers employ bots to track transactions and insert their own in front of high-value trades so that they can profit from price movements before the victim's transaction is made. Although current countermeasures like transaction batching and encrypted mempools mitigate these attacks to some extent, they are not perfect. More effective detection techniques are required to distinguish between real high-frequency trading and malicious front-running. A critical challenge is designing ML models capable of detecting faint patterns characteristic of front-running while avoiding false positives, promoting equitable trading in Ethereum-based DeFi platforms.

Maintaining the Scalability and Flexibility of Fraud Detection Systems

Fraud detection models need to adapt and change constantly to combat new fraud strategies used by fraudulent parties. Palaiokrassas *et al*. (2024), point out the challenge of applying ML-based fraud detection systems to novel DeFi protocols and changing attack patterns. Most fraud detection systems use pre-defined rules or static ML models, which get obsolete very soon as fraudsters advance their methods. Moreover, as the volume of Ethereum transactions increases, the currently available fraud detection solutions experience scalability issues and find it hard to handle massive volumes of data quickly. One of the central open problems is developing fraud detection mechanisms that are based on adaptive learning methods, including reinforcement learning and online learning, and which are computationally efficient. This problem will be key to maintaining strong security for large-scale Ethereum networks.

Methods

Data Preprocessing

Column management was an important part of the dataset cleaning process. The removal of any unnecessary columns, such as "Unnamed: 0", allowed for more efficient use of the data by consolidating multiple entries into one entry with relevant data. In addition, standardising column names throughout the preprocessing stage helped maintain consistency and create a more uniform result for analysis. Log transformations were applied to numerical attributes to reduce skewness, stabilise variance and create a more normally distributed dataset. All features were also normalized during preprocessing by scaling them to fall between the values of 0 and 1, as shown in the data preprocessing diagram (Figure 1), so that all features would have the same influence on the training of the model without having a disproportionate effect on those features that had a higher range of values.

Model Training

Algorithm Selection

The XGBClassifier was used as the learning algorithm because of its robust performance in binary classification problems. The classifier works based on a binary logistic objective, which is appropriate for the current problem. The training evaluation was performed using the log loss metric, a common measure for evaluating the performance of binary classifiers.

Hyperparameter Tuning

To evaluate the model's performance against a wide range of hyper-parameter configurations, a complete parameter search grid containing many of the key hyper-parameters used in building an optimizing model (i.e. max depth, the number of estimators, and learning rate) is created so that every combination of hyper-parameters will be thoroughly evaluated by GridSearchCV with Stratifsied K-Fold Cross Validation as shown in figure 2. By using an entirely stratified K-Fold Cross Validation technique, every fold is guaranteed to have the same class distribution as that of the original training data and will also allow multiple hyper-parameter combinations to be tested thoroughly without risking overfitting the training dataset.

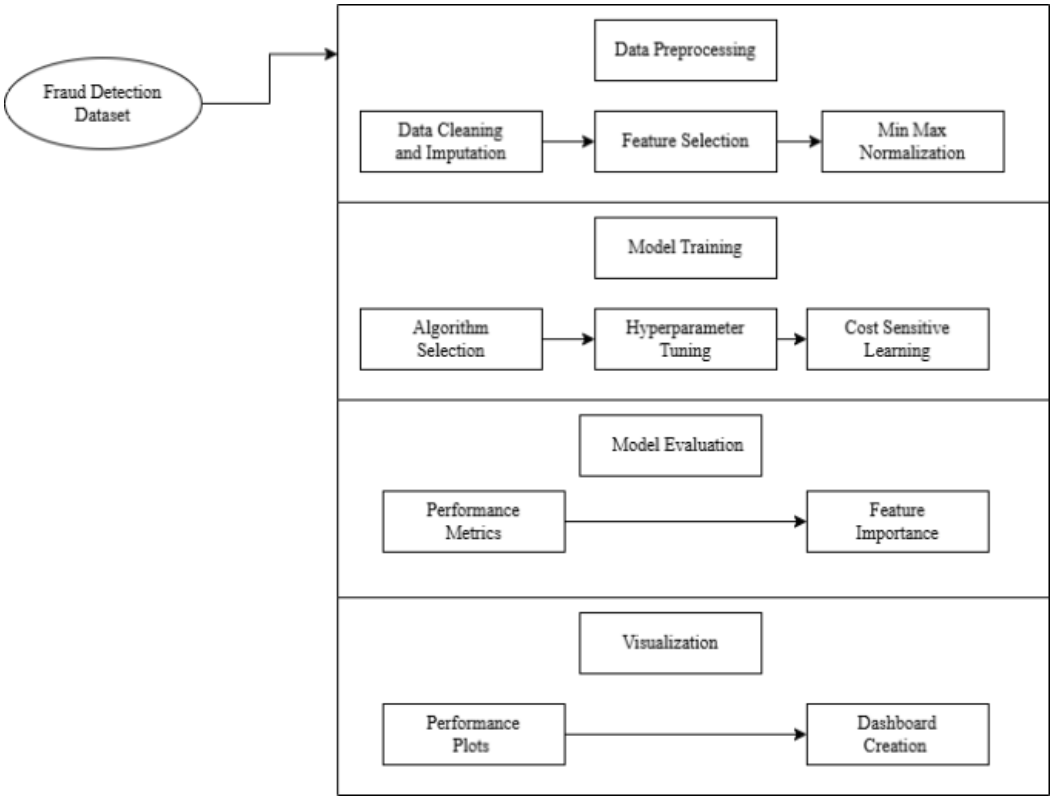

Fig. 1: Pipeline of data preprocessing, feature engineering, and XGBoost modelling

```
Fitting 5 folds for each of 27 candidates, totalling 135 fits
Best Hyperparameters: {'learning_rate': 0.05, 'max_depth': 7, 'n_estimators': 1000}
Best Training F1 Score: 0.9927413084461625
Test Accuracy: 0.9895004233700254
Test ROC-AUC: 0.9978959881375449
Test F1 Score: 0.9932740290735518
```

Fig. 2: Stratified K-fold Cross Validation

Model Evaluation

Performance Metrics

The various ways that can be used to evaluate how well a classifier was performing include: The F1 Score, an overall measure of the accuracy of the classifier, and Recap, a performance measure which takes into account both the accuracy of the classifier's Positive Predictions and the Recall of those Predictions combined with the overall how well the model does as a classifier. Finally, the ROC-AUC (Receiver Operating Characteristic Area Under Curve) was calculated to quantify the effectiveness of the model in classifying events into more than two Classes, in the event of which a confusion matrix is used for detailed analysis of the model's classification performance and can provide additional insight into the types of errors made and where those errors originated from.

Feature Importance

The most important parameters used to identify fraudulent activity are seen within Figure 3 and are critical for identifying unusual financial behaviours. The average minimum time difference between outgoing transaction attempts is an important parameter for identifying transaction patterns, and in most cases, fraud is characterized by rapid succession or unusual frequency of attempt(s) made over a short period. The largest value received from a transaction is also an important indicator and can highlight fraudulent behavior when an incoming

transaction greatly exceeds the normal limits for a given account. Another set of transaction patterns that indicate fraudulent activity is transfers of large amounts of Ether (ETH) using ERC20 tokens; the irregularities associated with transfers of tokens do tend to highlight fraudulent behavior in all aspects of Blockchain & Decentralized Finance (DeFi).

The model demonstrated in Figure 4 exhibits excellent performance with high accuracy, recall, and precision. It effectively minimizes both false positives and false negatives. Key diagnostic metrics confirm its strong predictive and diagnostic capabilities. Overall, the model is highly reliable for real-world deployment in critical tasks.

Visualization and Analysis

The ROC (Receiver Operating Characteristic) curve, as shown in figure 5, was plotted to visualize the balance between the true positive rate and the false positive rate over dynamic boundary values. The ROC graph illustrates the model's functionality to discriminate between classes. A score of 0.96 in AUC (Area Under the Curve) indicates excellent performance. The curve is well above the diagonal line, which signifies random guessing, showing that the model can easily distinguish fraudulent transactions from normal transactions. High AUC scores show good fraud detection with few false positives.

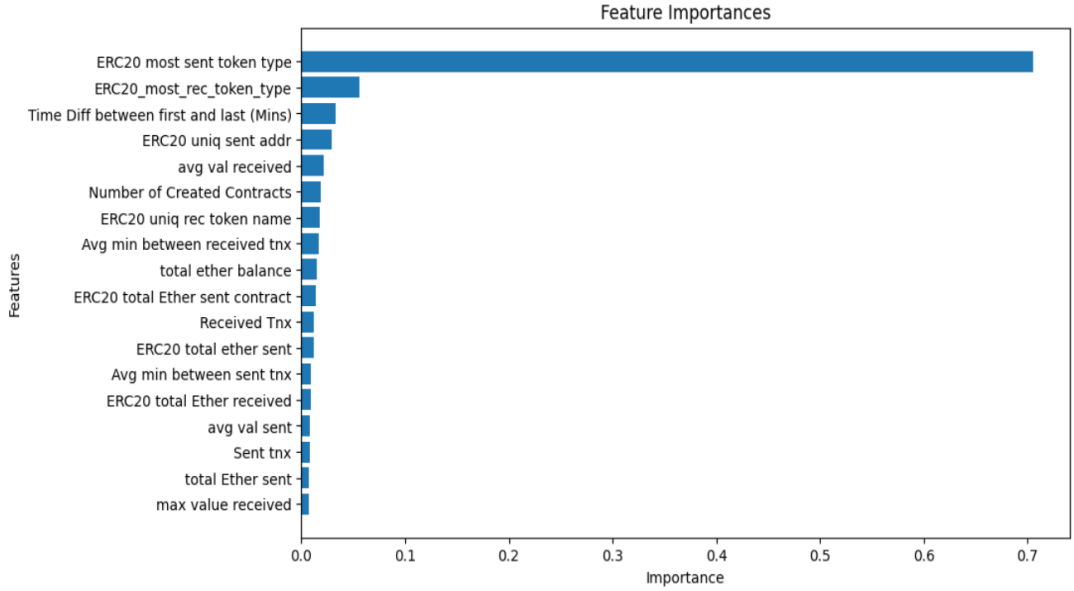

Fig. 3: Feature Importance

Model Diagnostic Statistics

Metric	Value
recall	0.9973856209150327
false_neg_rate	0.00261437908496732
false_pos_rate	0.03802281368821293
true_neg_rate	0.9619771863117871
positive_likelihood_ratio	26.23124183006536
negative_likelihood_ratio	0.0027177142266656333
precision	0.9891961970613656
false_omission_rate	0.009397024275646046
false_discovery_rate	0.010803802938634399
negative_predictive_value	0.990602975724354
markedness	0.9797991727857196
diagnostic_odds_ratio	9651.95
informedness	0.9593628072268197
prevalence_threshold	0.1633547993255626
prevalence	0.777307366638442
accuracy	0.9895004233700254

Fig. 4: Model Diagnostic Statistics showing comprehensive performance metrics of the classification model

Fig. 5: The ROC Curve

Through the plotting of curves, the analysis provided a thorough investigation into the threshold effect and determined how well the model was able to balance its precision with recall. As such, the analysis showed that, although recall was elevated, this did not reduce the precision of the model; thus, it provided a strong foundation for the ability to effectively detect fraud. Specifically, the model provided high levels of precision, which minimized the possibility of labelling legitimate transactions as fraudulent, while at the same time producing high levels of recall that reduced the likelihood that fraud would be missed. Therefore, in summary, the model is able to demonstrate a high degree of sensitivity and specificity, both of which are necessary for the provision of reliable and accurate fraud detection.

The Heatmap Visualization of a Confusion Matrix (as seen in Figure 6) provides one of the most visually appealing ways to view the performance of the classification model. It will provide a quick, easy to read view of how well this model is performing. The heatmap highlights all four areas of interest (TP, TN, FP and FN) so it will be easy to locate where errors are occurring. This heatmap shows the model classified 86% of all fraudulent transactions as TP, leaving only 14% of transactions as unidentified FPs, thus the threat of hidden fraud has been mitigated. It also shows that of the many legitimate transactions classified as FPs, a mere 0.10% were in error, indicating the model does not issue a high number of FPs. Additionally, the overall number of FN was very low and this is critical, as fraud detection scenarios can lead to dire consequences if FN occurs.

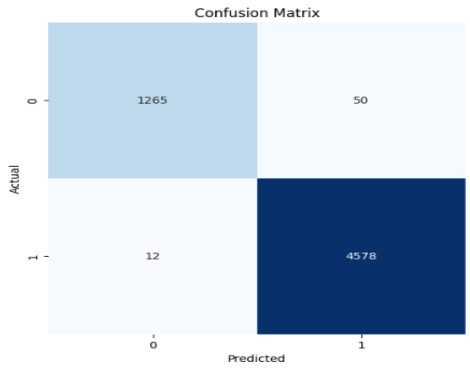

Fig. 6: Confusion Matrix

For improved stakeholder communication, an interactive dashboard was constructed with Plotly. The dashboard enabled users to interactively explore performance metrics, examine different visualizations, and develop a better intuitive understanding of the model's behaviour, as shown in Figure 9. When the user hovers over the graph in Figure 9, it displays the precise values of the features.

Functional Requirements

Data Collection and Preprocessing

The data collected included Ethereum transactions that were both acceptable and fraudulent. These data were sourced from blockchain explorers, proprietary transaction logs, and from security-focused organizations through a combination of all three. Each transaction is labelled accurately as either a fraudulent or legitimate transaction so that a supervised learning model can be trained and assessed effectively.

The preprocessing phase of preparing data includes finding all transactions with missing or null values for critical fields such as transaction amounts, sender and receiver addresses, and timestamps. The imputation method was used on transactions that had missing values when appropriate. Records that did not meet the required data quality were discarded from the dataset in order to maintain the validity and integrity of the data.

The validation of data was required to be done on an as needed basis in order to confirm the correctness and reliability of the data. The validation process involved checking for anomalies such as duplicate records and incorrect or inconsistent data entries which would negatively affect how well models perform. Another component of the validation process was the standardization of all fields (i.e., date and time, number of decimal points) so that the consistency of the various features within the dataset could be maintained.

Normalizing and scaling techniques were utilized on the data to allow for meaningful comparison of numerical features. Various scaling techniques (i.e., MinMax scaling or Z-score standardization) were applied to the data in order to ensure that all features were on the same scale so that features with large numerical ranges would not have a disproportionate effect on the learning process.

The feature engineering process was conducted to derive meaningful attributes from transaction data that could be used to describe fraudulent behavior. Some of the features derived from the raw transaction data that could effectively describe fraudulent activity included: frequency of transactions; transaction amount; transaction intervals; and network metrics which were designed to reflect the pattern of behavior and relationship of entities in the transaction network.

Fraud Detection

A variety of classification techniques have been tested to determine which are most effective at detecting fraud. Logistic Regression serves as a baseline linear approach when estimating the likelihood that any given transaction is fraudulent. The benefit of this type of algorithm is its intuitive nature and ease of interpretation in binary classification scenarios. To enhance the overall accuracy of classification, Random Forest employs an ensemble approach consisting of several decision trees, which address the potential for both overfitting and aspects of the data that are complex and non-linear. The K-Nearest Neighbors method uses the distance of a transaction to surrounding transactions in the feature space to classify it, offering a very intuitive distance-based process. However, the performance of K-Nearest Neighbors is generally poor in high-dimensional environments. AdaBoost is another boosting methodology which utilizes the concept of combining several weak learners to create one powerful learning model through the iterative application of errors from prior iterations. Lastly, XGBoost stands out as an efficient method for creating models that can accommodate large-scale datasets containing complex interactions between multiple features, thereby allowing for the efficient identification of fraudulent transactions.

Model Selection and Evaluation

Analyzing the model's performance included a set of metrics that would evaluate how well the model is able to accurately classify data based on the presence of an example class imbalance between fraud and non-fraud cases while also evaluating robustness across multiple metrics. The correctness of the positive predictions was used to measure the accuracy of fraud detection. In the fraud detection scenario, high values indicate that fraud was correctly identified and thus that false positives are minimized. Recall was used to evaluate the model's ability to identify fraudulent transactions out of all fraudulent transactions. This helped to reduce the number of false negatives. The F1 Score was calculated as the harmonic mean of correctness of positive predictions and recall, and serves as a balanced evaluation metric for imbalanced datasets. The model's overall accuracy was also measured through the as the degree of correctness, although this value is viewed with caution, given that this evaluation may be misleading when it is used to evaluate fraud detection systems that have large numbers of non-fraudulent transactions. The ROC-AUC metric was used, though it was optional, to assess the model's ability to differentiate fraudulent from non-fraudulent transactions across all thresholds.

After training multiple models, their performance was compared using the above metrics and the most efficient classifier (for example, XGBoost) was selected based on a balanced evaluation of these metrics. This step may involve further hyperparameter tuning to optimise the selected model.

Anomaly Detection

By comparing a person's transactions with others' transactions, we can identify any unusual activity, such as sudden increases in the amount of a transaction, unusually high numbers of transactions made by one person, and time-of-day transaction patterns that do not fit the typical patterns. Collaboratively detected anomalies were also found where groups of related accounts exhibited similar unusual activities, indicating that the accounts were being coordinated for fraud. In addition to using supervised classification techniques to identify fraudulent transactions, we also utilized unsupervised techniques such as clustering and density-based methods (like DBSCAN) to identify transactions that fall outside the overall distribution of data, helping us identify new or changing fraud patterns.

User Interaction

The dashboard has been designed with the administrator interface to allow the administrator to interact effectively with the fraud detection system using an intuitive and user-friendly method of use. The administrator is presented with current statistics and trends of suspicious activity within the dashboard; therefore, the administrator has the ability to monitor the performance of the fraud detection system and to quickly identify any potential new threats. In addition, the dashboard has a means for updating the fraud detection rules and decision thresholds; thus, the administrator can quickly react to new information as it becomes available or to any changing patterns of fraud. The dashboard supports the analysis of individual transactions with the ability to drill down into the details of each transaction, view transaction history, and view the relative importance of individual features or the detection of anomalies corresponding to specific predictions. In addition, the dashboard provides visualization tools, including ROC Curve (Figures 5), Confusion Matrix (Figure 6), Precision and Recall Curve, and Feature Importance Chart (Figure 9) that give clear understanding of how well models perform and provide reasons behind the predictions. This

will help the administrator with making better decisions and aiding interpretation of the information.

Detailed Methodology

Data Preprocessing

The dataset underwent six primary preprocessing methods. These are described as follows.

Data Reduction

Columns with redundant information, such as "Unnamed: 0," were removed to streamline the dataset and eliminate unnecessary details.

Labeling Consistency

Column labels were cleaned by removing extra spaces to ensure consistency and avoid errors during model training.

Target Feature Encoding

The target feature, "FLAG", which indicates fraudulent transactions, was encoded using LabelEncoder. This transformed categorical data into a numerical format, as shown in Equation (1):

$$\text{FLAGencoded}=\begin{cases}-1, if\ transaction\ is\ fraudulent \\ 0,\ otherwise\end{cases} \quad (1)$$

Addressing Feature Skewness

Features like "Total ether balance" and "Avg min between sent tnx" were log-transformed to resolve skewness.

The model's understanding of these features was enhanced by the normalization step using Equation (2):

$$Y' = \log(Y + 1) \quad (2)$$

Here Y is the original feature value and Y' is the transformed value.

Feature Scaling

MinMaxScaler was applied to scale all feature values to a range of [0,1]. Equation (3) was used for feature scaling:

$$Y_{scaled} = \frac{Y - Y_{min}}{Y_{max} - Y_{min}} \quad (3)$$

Here Y is the original feature value, Y_{min} is the maximum value of the feature, and Y_{max} is the maximum value of the feature.

Data Splitting

The dataset was split into training and testing subsets using a 70-30 ratio. Stratified sampling was employed to maintain the natural distribution of fraudulent and legitimate transactions in both subsets.

Model Development and Deployment

Model Training

The training process focused on selecting and optimizing the model to classify fraudulent Ethereum transactions.

Model Selection

An XGBoost classifier was selected for its high efficiency and strong performance, particularly with imbalanced datasets. The classifier was initialized using a binary logistic objective function, which refers to the model's assumption of how the output is generated. Here, we are predicting a binary outcome such as fraudulent (1) or non-fraudulent (0) transactions.

The probability that a given input Y results in a positive classification $a = 1$ is modelled using the sigmoid function:

$$P(a = 1|Y) = \frac{1}{1+e\text{^}-z} \quad (4)$$

This sigmoid function in Equation (4) is used to map the linear output of the model, denoted by z, to a probability between 0 and 1.

The term z is a linear combination of the input features Y, that produces a value that is passed through the sigmoid function to give a probability estimate:

$$z = w^T Y + b \quad (5)$$

Where w is the weight vector associated with the input features, Y is the input feature vector, and b is the bias term that shifts the decision boundary.

Equation (5) transforms the linear combination z into a probability between 0 and 1, allowing the model to interpret its output probabilistically for classification decisions.

Hyperparameter Tuning

Key hyperparameters, such as max_depth, n_estimators, and learning_rate, were optimized to balance model complexity, training time, and predictive performance.

Hyperparameter Optimization

GridSearchCV with 5-fold Stratified K-Fold cross-validation was used to determine the optimal set of hyperparameters.

The following hyperparameter grid was explored: max_depth ∈ {3, 5, 7}, n_estimators ∈ {100, 500, 1000}, and learning_rate ∈ {0.1, 0.05, 0.013}. This grid was

chosen to balance underfitting and overfitting while ensuring manageable training time.

The cross-validation error was optimized using Equation (6):

$$E_{cv} = \frac{1}{k}\sum_{k=1}^{k} E_k \quad (6)$$

Here E_k is the validation error on the k^{th} fold, k is the number of cross-validation folds (here, k = 5), and the summation collects the error from each fold.

Cost-Sensitive Learning

To address the issue of class imbalance in fraud detection, cost-sensitive learning was applied by assigning higher misclassification penalties to fraudulent transactions. The approach adopted reflects the business risk, where failing to detect fraud (false negative) is considered significantly more costly than flagging a legitimate transaction as fraud (false positive).

A cost matrix is defined in Equation (7):

$$Cost = \begin{bmatrix} 0 & C_{fn} \\ C_{fp} & 0 \end{bmatrix} \quad (7)$$

Here, C_{fn} is the cost of misclassifying a fraudulent transaction as legitimate, while C_{fp} is the cost of misclassifying a legitimate transaction as fraudulent.

In this study, C_{fn} was set to 5 and C_{fp} to 1, reflecting a 5:1 cost ratio. This ratio is justified as the financial and reputational loss from undetected fraud is typically far greater than the inconvenience of a false alert.

To integrate this cost-sensitive into training, the binary cross-entropy loss function in Equation (8) was modified using Class weights:

$$Loss = \sum_i w_i(a_i \log p_i) + (1 - a_i)\log(1 - p_i)) \quad (8)$$

Here, w_i represents the weight assigned to each class based on class distribution, a_i is the actual class label for instance i (either 0 or 1), p_i is the predicted probability for class 1 i.e., the model's output.

The loss function is used to train the classifier by penalizing incorrect predictions.

Final Model Training

After identifying the best hyperparameters, the model was trained on the scaled dataset. The XGBoost model follows an additive training approach as illustrated in Equation (9):

$$F(Y) = \sum_{m=1}^{M} am * hm(Y) \quad (9)$$

Here, $F(Y)$ is the final estimation for input Y, M is the total number of decision trees, hm(Y) represents individual decision trees, and am is the weight of each tree

XGBoost trains the model additively, one tree at a time. Each tree hm(Y) is added to correct the errors of the previous ensemble of trees. Over M iterations, the final model is an ensemble of these weighted trees. The weights am determined how much influence each tree has on the final prediction.

Enhanced Methodological Suggestions.

To improve the model's detection of relational and network-level fraud patterns, features were developed from graphs. PageRank and betweenness centrality scores for each address in the transaction network quantized how influential and intermediary the address was in the graph. Community detection labels were added as categorical features so that the model could better recognize fraud through multiple transactions as opposed to only through individual transaction attributes.

Anomaly detection was added as an additional feature to improve the model's ability to detect fraud by incorporating an Isolation Forest model to identify outliers of transactions. The unsupervised model produced an anomaly score and this was added as another input feature for the classifier to consider when evaluating a transaction to identify unusual or previously unseen behaviors associated with transactions from the overall data distribution.

Through the application of Bayesian optimization, model hyperparameters and settings that included cost sensitivity could be efficiently optimized. Optuna was used to optimize parameters, such as maximum depth of trees and learning rates, and to create a joint optimization with respect to the cost weights for class-specific false negatives and false positives. This provided an improved computational efficiency for searching through the hyperparameter space as compared to typical grid-based optimizing techniques.

In order to conduct performance evaluations of models that would be unbiased and reliable, nested cross-validation with a two-level validation process was used. The hyperparameter tuning and cost-weighting were done in the inner loop while the outer-loop was for estimating the performance of the model on unseen data. By separating tuning from evaluation, it helped to avoid having any information that could be leaked between the two processes, thus providing a stronger indication of how well the model would work in the real world.

Evaluation

The trained model's performance was evaluated using several metrics to assess its ability to correctly classify fraudulent transactions.

Performance Metrics

Key evaluation metrics included the following.

Degree of exactness i.e. the overall correctness of predictions was calculated using Equation (10):

$$Degree\ of\ Exactness = \frac{TP+TN}{TP+TN+FP+FN} \quad (10)$$

Using Equation (11), correctly identified positives out of all predicted positives were calculated:

$$Correctness\ of\ positive\ predictions\ = \frac{TP}{TP+FP} \quad (11)$$

Equation (12) correctly identifies positives out of all actual positives:

$$Recall\ = \frac{TP}{TP+FN} \quad (12)$$

Equation (13) illustrates the harmonic mean of precision and recall i.e. *F*1 score:

$$F1 = \frac{2 \times Precision \times Recall}{Precision+Recall} \quad (13)$$

Here, TP (True Positives) are the fraudulent transactions, TN (True Negatives) are the transactions identified as non-fraudulent correctly, FP (False Positives) are the genuine transactions incorrectly predicted as fraudulent, and FN (False Negatives) are the fraudulent transactions incorrectly predicted as genuine.

Practical Considerations and Future Outlook

The EMSCAD dataset used in this study is one of the most comprehensive public datasets available for Ethereum fraud detection. However, it is important to know that this dataset may not capture all types of fraud (e.g., sophisticated smart contract exploits or flash loan attacks). Future work could include live data feeds and API accessible blockchain platforms to improve real-time applicability.

Scalability Considerations: While the proposed system performs well on a static dataset, real-time deployment would require optimizations for latency and throughput. This includes using streaming data architectures (e.g., Apache Kafka), parallelizing feature extraction with Spark, and compressing model inference with ONNX or TensorRT.

These considerations lay the groundwork for transitioning the proposed system from a research prototype to a production-ready, ethically responsible fraud detection solution.

Results and Discussion

As a way of determining the effectiveness of the proposed fraud detection technique, we compared the performance of an XGBoost classifier with cost-sensitive learning and without cost-sensitive learning. The models were determined by the degree of exactness with which they precisely detect fake and genuine Ethereum transactions. Performance measures such as confusion matrices were employed to calculate the proportion of false negatives and false positives.

Feature importance: Analysis was conducted to assess the most striking features of deceptive conduct. The results emphasize progress achieved by cost-saving strategies and offer insights into important transaction attributes for fraud detection.

XGBoost With Cost-Sensitive Learning

Examining the outputs in Fig. 7 provided confirmation that this was a successful model with regard to correctly identifying and incorrectly classifying transactions. In total, 1271 transactions correctly identified as legitimate (non-fraud) i.e., true negatives and 44 transactions classified as fraudulent while being legitimate i.e., false positives. Conversely, 15 fraudulent transactions were misidentified as legitimate i.e., false negatives. As per the total fraudulent transactions, this resulted in a total of 4575 correct classifications (true positives) of fraudulent transactions. During the entire assessment period, this distribution supports this model's ability to classify transactions correctly in a limited capacity with low error rates.

XGBoost Without Cost-Sensitive Learning

The model's classification accuracy is depicted in Figure 8 through a display of the classification results for both correctly classified and misclassified transactions. Out of a total of 1,264 non-fraudulent transactions classified correctly, 51 legitimate transactions were misclassified as fraud (false positive). Of the 14 fraudulent transactions classified incorrectly (false negative), 4,576 were accurately identified as fraud (true positive). The overall results indicate a high level of accuracy with regard to detection of fraudulent activity and a low level of misclassification in distinguishing between legitimate and fraudulent transactions.

The cost-sensitive model improved in lowering the false positives by reducing them from 51 to 44, a decrease by 13.7%. This is especially useful in practical applications, where reducing false positives can increase user confidence and avoid unnecessary manual checks. The cost-sensitive model did have a minor increase in false negatives, from 14 to 15, which means that one more fraudulent transaction was not detected. Despite this compromise, the overall classification accuracy was better, especially in dealing with class imbalances, which is a major issue in fraud detection. Table 1 summarizes the precision, recall, and F1-score for both cost-sensitive and baseline models.

From a business point of view, false positive reduction is important because it minimizes the operational load on fraud investigators and allows for more effective resource deployment.

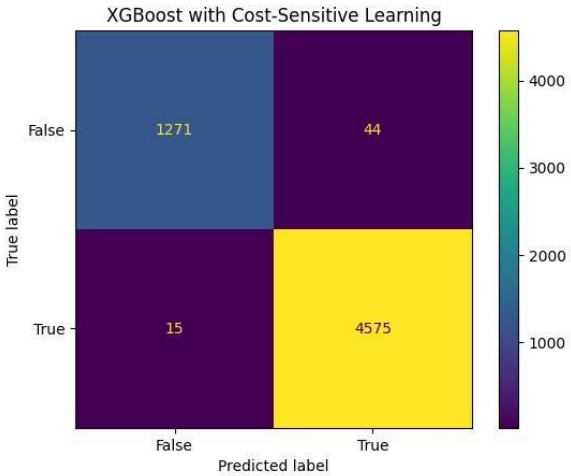

Fig 7: XGBoost with Cost-Sensitive Learning

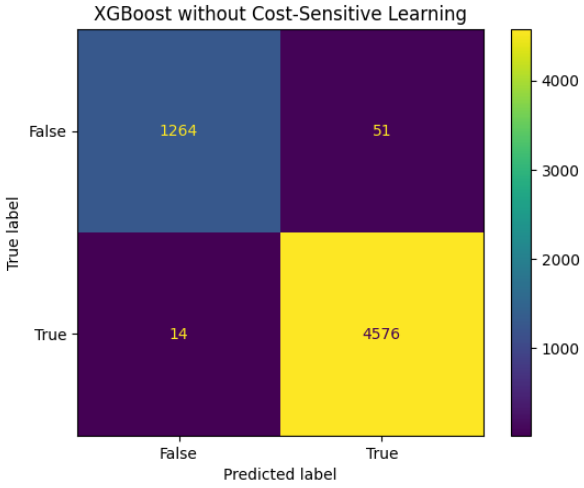

Fig 8: XGBoost without Cost-Sensitive Learning

Table 1: Comparative table with precision, recall, and F1-score for both cost-sensitive and baseline models

Model	Precision	Recall	F1 Score
XGBoost (Base)	98.90%	99.69%	99.29%
XGBoost (Cost-Sensitive)	99.05%	99.67%	99.36%

Feature Importance Analysis: Of the features that were tested, 'ERC20 most sent token type' was a prominent predictor in the fraud detection model, as illustrated in Figure 9. Its high importance indicates a strong correlation with the target variable ('FLAG'), which labels transactions as fraudulent or genuine. ERC20 i.e. Ethereum Request for Comment 20. It is an extensively used protocol for fungible tokens on the Ethereum blockchain to provide compatibility with wallets, exchanges, and smart contracts. ERC20 is widely employed for financial transaction purposes such as payments, capital raising, and decentralized applications. Due to extensive usage, a particular type of token might get disproportionately linked to fraudulent transactions and hence become a key component of fraud detection systems.

Certain types of ERC20 tokens can be common in fraudulent transactions, and thus, this can be a good predictive sign of suspicious activity. Furthermore, the pattern of token types could differ markedly between non-flagged and flagged transactions, with some tokens being overrepresented among fraudulent cases. This feature does not necessarily perform independently but may complement transaction volume, frequency, and balance in order to detect fraud better. XGBoost picks up such interactions, and its contribution to classification increases. Because label encoding was used on this feature, the numerical representation may have introduced patterns that were used by the model for improved classification. The transformation can have allowed the model to identify differences between various token types more effectively. Infrequently occurring token types that often occur in suspicious transactions can serve as implicit signals of fraud, enabling the model to detect high-risk activity.

Conclusion

This work demonstrates the performance of blending cost-sensitive learning into the XGBoost Classifier for identifying illicit transactions on the Ethereum blockchain. To improve the real-world fraud detection systems, the model adopted class imbalance strategies, which resulted in a significant reduction in false positives. The cost-sensitive approach had a slight increase in false negatives; however, the overall performance reflected a balanced and improved classification capability. In addition, feature importance analysis explored that specific transaction attributes, especially ERC20 token types, were crucial in distinguishing fraudulent behavior.

The key contributions include a 13.7% reduction in false positives through cost-sensitive XGBoost tuning, aptly identifying ERC20 token type as a novel, high-impact feature in fraud classification and a prototype dashboard for real-time on-chain monitoring, enhancing practical deployability.

It can be concluded that the cost-sensitive XGBoost approach, integrated with in-depth feature extraction, offers a robust framework for blockchain-based fraud detection, enhancing both accuracy and practical relevance in security applications.

Acknowledgment

The authors sincerely acknowledge B.M.S. College of Engineering (BMSCE), Bengaluru, and the

Department of Machine Learning for providing the necessary facilities, resources, and continuous support to carry out this work.

Funding Information

This research received no external funding.

Authors Contributions

Supriya P: Conceptualization, research design, formulation of research objectives, methodological design, project direction, system architecture insights, study development and implementation, manuscript drafting, critical revision, final editing and refinement.

Rubah Sheriff: Abstract drafting, methodology development, proposed architecture design, data acquisition, data collection oversight, data pre-processing, data cleaning, manuscript structuring, methodology articulation.

Shreya Padaki: Introduction, research gap identification, literature synthesis, model training design, algorithm selection, model performance evaluation.

Suchi V Yadav: Literature review, state-of-the-art survey, synthesis of research, critical analysis of strengths and limitations, model evaluation design, performance metric implementation (F1 score, ROC-AUC, confusion matrix), experimental validation.

Varsha Thakur: Analysis of key open issues, data visualization, transaction distribution graphs, exploratory data analysis (EDA), feature selection guidance, visualization clarity, manuscript documentation, research quality enhancement.

Authors Contributions

Each author made substantial contributions to conception, design, data collection, analysis, implementation, drafting, and critically revised the manuscript. All five authors approved the final version and agreed to be accountable for all aspects of the work.

Data Availability Statement

The data used in this study are publicly available.

Conflicts of interest

The authors declare no conflict of interest.

References

Ashfaq, T., Khalid, R., Yahaya, A. S., Aslam, S., Azar, A. T., Alsafari, S., & Hameed, I. A. (2022). A Machine Learning and Blockchain Based Efficient Fraud Detection Mechanism. *Sensors*, *22*(19), 7162. https://doi.org/10.3390/s22197162

Aziz, R. M., Baluch, M. F., Patel, S., & Ganie, A. H. (2022). LGBM: Detection of illicit accounts approach for Ethereum fraud detection. International Journal of Information Technology. *International Journal of Information Technology*, *14*(7), 3321–3331. https://doi.org/10.1007/s41870-022-00864-6

Ding, Z., Shi, J., Li, Q., & Cao, J. (2024). Effective Illicit Account Detection on Large Cryptocurrency MultiGraphs. *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, 457–466. https://doi.org/10.1145/3627673.3679707

Dutta, A., Voumik, L. C., Ramamoorthy, A., Ray, S., & Raihan, A. (2023). Predicting Cryptocurrency Fraud Using ChaosNet: The Ethereum Manifestation. *Journal of Risk and Financial Management*, *16*(4), 216. https://doi.org/10.3390/jrfm16040216

Elmougy, Y., & Liu, L. (2023). Demystifying Fraudulent Transactions and Illicit Nodes in the Bitcoin Network for Financial Forensics. *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 3979–3990. https://doi.org/10.1145/3580305.3599803

Elmougy, Y., & Manzi, O. (2021). Anomaly Detection on Bitcoin, Ethereum Networks Using GPU-accelerated Machine Learning Methods. *2021 31st International Conference on Computer Theory and Applications (ICCTA)*, 166–171. https://doi.org/10.1109/iccta54562.2021.9916625

Farrugia, S., Ellul, J., & Azzopardi, G. (2020). Detection of illicit accounts over the Ethereum blockchain. *Expert Systems with Applications*, *150*, 113318. https://doi.org/10.1016/j.eswa.2020.113318

Giantsidi, S., & Tarantola, C. (2024). Deep learning for financial forecasting: A review of recent advancements. *SSRN (Social Science Research Network)*, 81.

Hu, H., Bai, Q., & Xu, Y. (2022). SCSGuard: Deep Scam Detection for Ethereum Smart Contracts. *IEEE Xplore*, 1–6. https://doi.org/10.1109/infocomwkshps54753.2022.9798296

Hu, J., Cao, M., Zhang, X., Zhang, X., & Zhu, Y. (2023). Temporal Weighted Heterogeneous Multigraph Embedding for Ethereum Phishing Scams Detection. *2023 26th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, 1208–1213. https://doi.org/10.1109/cscwd57460.2023.10152679

Ibrahim, R. F., Mohammad Elian, A., & Ababneh, M. (2021). Illicit Account Detection in the Ethereum Blockchain Using Machine Learning. *2021 International Conference on Information Technology (ICIT)*, 488–493. https://doi.org/10.1109/icit52682.2021.9491653

Islam, M. S., Bashir, M., Rahman, S., Al Montaser, M. A., Bortty, J. C., Nishan, A., & Haque, M. R. (2025). Machine Learning-Based Cryptocurrency Prediction: Enhancing Market Forecasting with Advanced Predictive Models. *Journal of Ecohumanism*, *4*(2), 2498–6663. https://doi.org/10.62754/joe.v4i2.6663

Jiang, W., Yin, P., & Zhu, W. (2024). *Cryptocurrency Transaction Fraud Detection Based on Imbalanced Classification with Interpretable Analysis*. 386–398. https://doi.org/10.1007/978-3-031-60260-3_32

Kanezashi, H., Suzumura, T., Liu, X., & Hirofuchi, T. (2022). *Ethereum fraud detection with heterogeneous graph neural networks*.

Kerr, D. S., Loveland, K. A., Smith, K. T., & Smith, L. M. (2023). Cryptocurrency Risks, Fraud Cases, and Financial Performance. *Risks*, *11*(3), 51. https://doi.org/10.3390/risks11030051

Krishna, M. V., & Praveenchandar, J. (2022). Comparative Analysis of Credit Card Fraud Detection using Logistic regression with Random Forest towards an Increase in Accuracy of Prediction. *IEEE Xplore*, 1097–1101. https://doi.org/10.1109/icecaa55415.2022.9936488

Osterrieder, J., Chan, S., Chu, J., & Zhang, Y. (2023). A Primer on Anomaly and Fraud Detection in Blockchain Networks. *SSRN Electronic Journal*, 24. https://doi.org/10.2139/ssrn.4317520

Palaiokrassas, G., Scherrers, S., Ofeidis, I., & Tassiulas, L. (2024). Leveraging Machine Learning For Multichain DeFi Fraud Detection. *2024 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, 678–680. https://doi.org/10.1109/icbc59979.2024.10634350

Sallam, A., H. Rassem, T., Abdu, H., Abdulkareem, H., Saif, N., & Abdullah, S. (2022). Fraudulent Account Detection in the Ethereum's Network Using Various Machine Learning Techniques. *International Journal of Software Engineering and Computer Systems*, *8*(2), 43–50. https://doi.org/10.15282/ijsecs.8.2.2022.5.0102

Sultana, S., Rahman, Md. S., & Afroj, M. (2023). An efficient fraud detection mechanism based on machine learning and blockchain technology. *2023 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)*, 162–168. https://doi.org/10.1109/3ict60104.2023.10391306

Sureshbhai, P. N., Bhattacharya, P., & Tanwar, S. (2020). KaRuNa: A Blockchain-Based Sentiment Analysis Framework for Fraud Cryptocurrency Schemes. *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*, 1–6. https://doi.org/10.1109/iccworkshops49005.2020.9145151

Taher, S. Sh., Ameen, S. Y., & Ahmed, J. A. (2024). Advanced Fraud Detection in Blockchain Transactions: An Ensemble Learning and Explainable AI Approach. *Engineering, Technology & Applied Science Research*, *14*(1), 12822–12830. https://doi.org/10.48084/etasr.6641

Tripathy, N., Kumar Balabantaray, S., Parida, S., & Nayak, S. K. (2024). Cryptocurrency fraud detection through classification techniques. *International Journal of Electrical and Computer Engineering (IJECE)*, *14*(3), 2918–2926. https://doi.org/10.11591/ijece.v14i3.pp2918-2926

Venkatesh Prasath, S., & Prabhu Kavin, B. (2024). Fraud detection and prevention in Ethereum transactions. *International Research Journal of Engineering and Technology*, *11*(5), 118.

Walavalkar, P., Dasrapuria, A., Sarda, M., & Dmello, L. (2024). A Token-based Approach to Detect Fraud in Ethereum Transactions. *International Journal for Research in Applied Science and Engineering Technology*, *12*(4), 45–98. https://doi.org/10.22214/ijraset.2024.59690

Zhang, Y., Liu, P., Wang, G., Li, P., Gu, W., Chen, H., Liu, X., & Zhu, J. (2024). FRAD: Front-Running Attacks Detection on Ethereum Using Ternary Classification Model. *Springer Nature Link*, *2034*, 63–75. https://doi.org/10.1007/978-981-97-1274-8_5