VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

CHENNAI

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING [SCOPE]

J COMPONENT REVIEW-III

Course Title: REAL TIME ANALYTICS

Course code: CSE3506

Topic: Ethereum Transaction Fraud Detection

Faculty: Dr. N.M Elango

Team Members:

Gaurav Dwivedi -20MIA1037

Rohit Choudhary -20MIA1069

Ethereum Transaction Fraud Detection

Abstract

Blockchain technology, particularly Ethereum, has gained immense popularity for its decentralized and immutable nature, revolutionizing various industries including finance, supply chain, and healthcare. However, with the rise in usage, fraudulent activities within the Ethereum network have become a pressing concern. Detecting fraudulent transactions is crucial for maintaining the integrity and trustworthiness of the Ethereum ecosystem. Since 2021, more than 46,000 people lost over $1 billion to cryptocurrency scams, nearly 60 times more compared to 2018.1 The Federal Trade Commission (FTC) found that the top cryptocurrencies used to pay scammers were Bitcoin (70%), Tether (10%) and Ethereum (9%).

Especially, with the most recent incident with FTX, a crypto exchange which misused more than $1 billion of client's funds, it becomes ever more important to stay vigilant when navigating through the cryptocurrency world.

To enforce deterrence against fraudulent scams, we used supervised machine learning techniques such as Logistic Regression, Naive Bayes, SVM, XG-Boost, LightGBM, MLP, Tabnet and Stacking to detect and predict fraudulent Ethereum accounts.

This would add business value by enhancing fraudulent account detection features on crypto exchanges and crypto wallets, enabling people to navigate confidently through the cryptocurrency world and safeguard their personal assets. We set an objective to achieve more than 90% F1 score for machine learning models in predicting fraudulent accounts on the Ethereum blockchain.

Introduction

The proliferation of blockchain technology and cryptocurrencies has ushered in an era of decentralized and transparent financial transactions. However, with the growth of these digital ecosystems, there has been a parallel surge in the occurrence of transaction fraud.

Ensuring the integrity and security of blockchain networks is paramount to maintaining trust and stability. The core objective of this project is to develop a sophisticated transaction fraud detection system tailored to Ethereum accounts. By deploying a blend of data preprocessing techniques and a comprehensive range of classification models, the project seeks to provide a robust solution for identifying and thwarting fraudulent activities within blockchain networks.

Blockchain technology, most notably exemplified by Ethereum, has fundamentally transformed financial transactions and asset transfers by eliminating the need for intermediaries. These networks have, however, become enticing targets for malicious actors seeking to exploit vulnerabilities. The rise of fraudulent activities such as unauthorized fund transfers, suspicious smart contract interactions, and identity spoofing necessitates the development of a dedicated system capable of detecting these illicit actions.

We delve into the intricacies of data preprocessing, essential for transforming raw transaction data into a format suitable for machine learning analysis. The pivotal role of this project is to contribute not only to the security of blockchain networks but also to the broader discourse surrounding cryptocurrency and blockchain technology. The results and insights derived from this endeavor inform our understanding of the evolving landscape of transaction fraud and advance the domain of fraud detection within blockchain ecosystems.

Dataset collection

There are 2 data sources :

Kaggle

The Kaggle dataset is downloaded from
https://www.kaggle.com/datasets/vagifa/ethereum-frauddetection-dataset

EtherScan

Data are mined from Etherscan from
https://etherscan.io/accounts/label/phish-hack

We started with a Kaggle dataset of 9841 observations. Each observation is a unique Ethereum account, with each variable being an
aggregate statistic over all transactions performed by that unique account, such as total Ether value received or average time between transactions.

The data also distinguishes between account-to-account transactions and account-to-smart contract transactions. However, the dataset was highly imbalanced, with only 2179 out of 9841 (22.14%) being marked as fraud.

To address the imbalance, we leveraged an API provided by Ether-scan, a "Block Explorer and Analytics Platform for Ethereum". address on the Ethereum blockchain. As a result, the number of fraudulent accounts in our dataset climbed to 4339 observations, making the combined dataset less imbalanced (45.97% fraud).

Literature Survey

[1.1] A Survey of Blockchain Security Issues and Challenges

Lin and Liao's comprehensive survey offers an insightful examination of the evolving landscape of blockchain technology, shedding light on the multifaceted security concerns and challenges that surround it. The paper delves into critical aspects such as consensus algorithms, cryptographic techniques, and smart contract security, offering a holistic perspective on the security vulnerabilities inherent in blockchain ecosystems. Through a thorough exploration of potential threats, this survey equips researchers and practitioners with a nuanced understanding of the ever-expanding domain of blockchain security. It serves as a valuable resource for addressing these security concerns and developing effective countermeasures.

[1.2] A Survey of Blockchain from a Security Perspective

Dasgupta, Shrein, and Gupta's survey delves into the multifaceted realm of blockchain technology, with a specific focus on security considerations.

The paper offers an extensive overview of the security implications inherent in blockchain systems, encompassing topics such as cryptographic techniques, consensus mechanisms, and smart contract vulnerabilities. By elucidating the potential security threats and presenting a comprehensive view of proactive security measures, this survey equips readers with a deep understanding of the complex interplay between blockchain and security.

It provides an invaluable resource for researchers, practitioners, and policymakers seeking to navigate the security landscape in an era of blockchain-driven innovation.

[1.3] Fraud detection system: A survey

The paper authored by Abdallah, Maarof, and Zainal provides an in-depth survey of fraud detection systems, offering insights into the multifaceted domain of identifying and preventing fraudulent activities. The survey comprehensively examines the strategies, methodologies, and technologies deployed in the detection of fraud across various domains. It highlights the significance of robust fraud detection systems in maintaining the integrity of financial transactions and digital ecosystems. This survey acts as a pivotal resource for researchers and professionals seeking to enhance their understanding of the evolving landscape of fraud detection, offering a holistic perspective on the challenges and solutions associated with this critical domain.

[1.4] Fraud Detection and Verification System for Online Transactions: A Brief Overview

Chauhan and Tekta's paper offers a concise yet informative overview of fraud detection and verification systems tailored for online transactions. The work sheds light on the critical role of such systems in ensuring the security and integrity of online financial interactions. It explores various techniques and methods employed for fraud prevention, including anomaly detection, authentication mechanisms, and real-time monitoring. This overview serves as a valuable introduction to the domain of online transaction security, providing insights into the challenges and strategies associated with combating fraud in the digital realm.

[1.5] A Data Mining Based System for Transaction Fraud Detection

The paper authored by Deng, Huang, Zhang, and Xu presents a data mining-based system designed for transaction fraud detection. It offers a novel approach to addressing the critical issue of fraud in financial transactions. The work explores various data mining techniques, including classification and anomaly detection, as well as the utilization of machine learning algorithms. By leveraging data mining, this system enhances the capability to identify suspicious transactions and mitigate fraudulent activities. It serves as a noteworthy contribution to the field of transaction fraud detection, providing insights into the evolving strategies and methodologies for safeguarding financial interactions.

[1.6] Graph Neural Network for Ethereum Fraud Detection

Tan, Tan, Zhang, and Li's paper introduces an innovative approach to fraud detection in the context of Ethereum using Graph Neural Networks (GNNs). The study underscores the importance of leveraging graph-based data representations to enhance the identification of fraudulent activities within the Ethereum blockchain. By harnessing GNNs, this work extends the capabilities of fraud detection systems to capture complex relationships among transactions, addresses, and smart contracts. It explores the potential of GNNs to uncover patterns and anomalies in transaction data, thereby fortifying the security of Ethereum's decentralized financial ecosystem.

[1.7] Efficient Fraud Detection in Ethereum Blockchain through Machine Learning and Deep Learning Approaches

In this research authors have taken ethereum dataset then after data preprocessing the data was splitted into training and testing dataset and then
used to train different Machine Learning Models

and then the trained model is used to classify the
transaction data into fraud and non-fraud transactions.

High Precision Rate: The proposed model boasts a precision rate of 97.16%, marking a substantial leap in fraudulent transaction detection on the Ethereum blockchain in comparison to prevailing methodologies.

Enhanced Fraud Detection: The model is designed to identify anomalies in Ethereum transactions, flagging them as fraudulent or legitimate.

Improved Security: The model can help prevent significant monetary losses due to fraudulent activities in the Ethereum blockchain.

[1.8] A Machine Learning and Blockchain Based Efficient Fraud Detection Mechanism

Machine learning algorithms (XG boost and random forest) for transaction classification.

Integration of blockchain technology with machine learning algorithms for fraud detection

Proposedatt acker model to protect system from attacks and vulnerabilities

This study utilizes machine learning algorithms (XG-Boost and random forest) for transaction classification and prediction, enhancing fraud detection in the Bitcoin network.

And this study provides a secure fraud detection model that can help mitigate the increasing number of frauds in the financial sector.

[1.9] LGBM: A machine learning approach for Ethereum fraud detection

Ethereum is a cryptocurrency-transfer system based on a blockchain network.

- Teng et al. proposed an approach using data slicing and LSTM.
- Steven et al. used XG-Boost to detect malicious accounts.
- Qi et al. developed a model to detect phishing frauds.
- Ajay et al. used machine learning techniques to detect anomalies.

The study uses a customized machine learning approach using gradient boosting algorithms that can detect anomalous transactions with high accuracy without being vulnerable to overfitting.

[1.10] Exploiting Blockchain Data to Detect Smart Ponzi Schemes on Ethereum

The document discusses the collection of ground truth data, feature extraction from transaction history and bytecode, and the extraction of code features from operation codes. It also presents the classification model used for detecting smart Ponzi schemes.

The advantages of the proposed approach include the use of publicly accessible blockchain data for fraud detection, the development of a better classification model, and the identification of new account features for improved classification.

[1.11] Fraud Detection: A Review on Blockchain

Focus on rating fraud and different models of fraudulent ratter behavior

Blockchain systems good at avoiding objective information fraud.

- Blockchain compliance mechanism
- Supervised learning techniques
- Anomaly detection.
- Use of trimmed k-means algorithm.
- Exploiting the immutability of blockchain for security and transparency.

The study include improved data security, privacy, and integrity, as well as the potential for reducing the time and cost of verifying job history. The disadvantages involve the reliance on data producers, the need for transaction fees, and the challenge of preventing all types of fraud.

[1.12] Predicting Cryptocurrency Fraud Using Chaos Net: The Ethereum Manifestation

ML classifier in conjunction with

- Chaos Feature Extractor
- AdaBoost algorithm
- Macro F1 score evaluation
- GLS neurons in Chaos Net

Chaos Net is an Artificial Neural Network (ANN) constructed using Generalized Luröth Series maps, which allows for classification problems with cutting-edge performance and lower training sample count.

Chaos Net utilizes the chaotic GLS neurons' property of topological transitivity, which enables it to solve complex classification tasks on par with or better than standard ANNs.

Chaos Net combines the best traits of networks composed of biological neurons, derived from the strong chaotic activity of individual **neurons, to** achieve better results than generic ML algorithms.

It is very difficult to train the Chaos Net model, on unpredictable and noisy data, in neural networks and this make it challenging for getting the expected results from the neural networks model. The survey begins by discussing the significance of fraud detection in the cryptocurrency ecosystem, highlighting the financial and reputational risks associated with fraudulent activities. It then provides an overview of Ethereum, emphasizing its smart contract functionality and its role as a breeding ground for various forms of fraud.

Subsequently, the survey delves into the theoretical foundations of chaos theory and its relevance to predictive modeling. Chaos Net, as a derivative of chaos theory, leverages nonlinear dynamics and feedback mechanisms to capture complex interactions within cryptocurrency transaction networks, enabling the detection of anomalous patterns associated with fraudulent behavior.

The survey reviews existing literature on cryptocurrency fraud detection techniques, including traditional machine learning approaches, anomaly detection methods, and blockchain analysis tools. It compares and contrasts these methods with Chaos Net, highlighting the unique advantages of Chaos Net in capturing nonlinear dynamics and emergent behaviors inherent in cryptocurrency transactions.

[1.13] Unsupervised Learning for Robust Bitcoin Fraud Detection

In this study the authors used the trimmed k-means algorithm for fraud detection in Bitcoin transactions. This algorithm is capable of simultaneous clustering of objects and fraud detection in a multivariate setup.

The majority of the transactions on the network are assumed to be legitimate, with at most only 1 being fraudulent, as per the anomaly detection techniques used in the study. The study on "Unsupervised Learning for Robust Bitcoin Fraud Detection" presents key advantages, including improved fraud detection rates compared to existing methods. It employs unsupervised learning, specifically trimmed k-means, for anomaly identification in the Bitcoin network. This approach offers a valuable solution to combat fraud and enhance the security of financial transactions within the blockchain ecosystem.

[1.14] Investigating the impact of structural and temporal behaviors in Ethereum phishing users detection

The paper proposes methods for detecting Ethereum phishing scams, including sample selection, graph abstraction, feature aggregation, and classification. The methods include PDTGA, which uses a temporal graph attention network, and HTSGCN, which is based on heterogeneous transaction subnets.

The paper also discusses the classification of transaction network subgraphs using the EGAT procedure.

Investigates the effects of network architectural features and temporal aspects.

Uses traditional machine learning algorithms to evaluate the model Proposed features identify phishing accounts efficiently.
Dataset does not include transaction fees, gas price, and confirmation time. Highly skewed dataset with low number of phishing addresses. Class imbalance addressed using SMOTE over-sampling and Random under-sampling. Enriched feature set with graph-based data modeling. Lack of awareness leading to new users falling for phishing scams.

[1.15] Systematic Review of Security Vulnerabilities in Ethereum Blockchain Smart Contract

The paper discusses Ethereum smart contract security vulnerabilities, detection tools, real-life attacks, and preventive mechanisms.
The systematic review approach allows for a comprehensive analysis of security vulnerabilities in Ethereum blockchain smart contracts, ensuring that all relevant information is considered .Comparisons among Ethereum smart contract analysis tools provide valuable insights into the strengths and weaknesses of different tools, aiding researchers and practitioners in selecting the most suitable tool for their needs.
The limitations of the Ethereum Virtual Machine (EVM), such as the restricted stack size and gas consumption limit, are mentioned as factors that can lead to vulnerabilities in smart contracts. Furthermore, we examine the root causes and underlying mechanisms of each vulnerability type, providing insights into the factors contributing to their exploitation. This analysis aids in understanding the complexity of smart contract security and informs developers and auditors about potential pitfalls to avoid during contract development and auditing processes.

Moreover, the review explores existing methodologies and tools designed for detecting, analyzing, and mitigating smart contract vulnerabilities. We discuss static and dynamic analysis techniques, formal verification methods, and automated vulnerability scanners, highlighting their strengths and limitations in the context of Ethereum smart contract security.
In addition, we discuss real-world incidents and case studies where smart contract vulnerabilities have been exploited, leading to significant financial losses and reputational damage. By examining these case studies, we underscore the importance of proactive security measures and continuous auditing practices in safeguarding smart contracts against potential threats.

Module Decomposition

1. **Data Acquisition**

   Ethereum Blockchain: Access historical and real-time transaction data (e.g., addresses, timestamps, values, smart contract interactions) through Ethereum Node or APIs like Infura or Alchemy.

   External Data Sources: Integrate data from other relevant sources like IP geolocation, wallet reputation scoring, and known scam lists.

2. **Data Preprocessing & Engineering:**

   Extract features: Transform raw data into meaningful features (e.g., transaction size, frequency, time of day, contract type, wallet age).

   Handling missing values: Impute missing data or remove incomplete entries. Feature scaling: Normalize features for better model performance.

3. **Model Training & Evaluation:**

   Machine Learning Model: Choose an appropriate model like:

   Supervised learning: Logistic regression, Random Forest and (fraudulent/non-fraudulent).

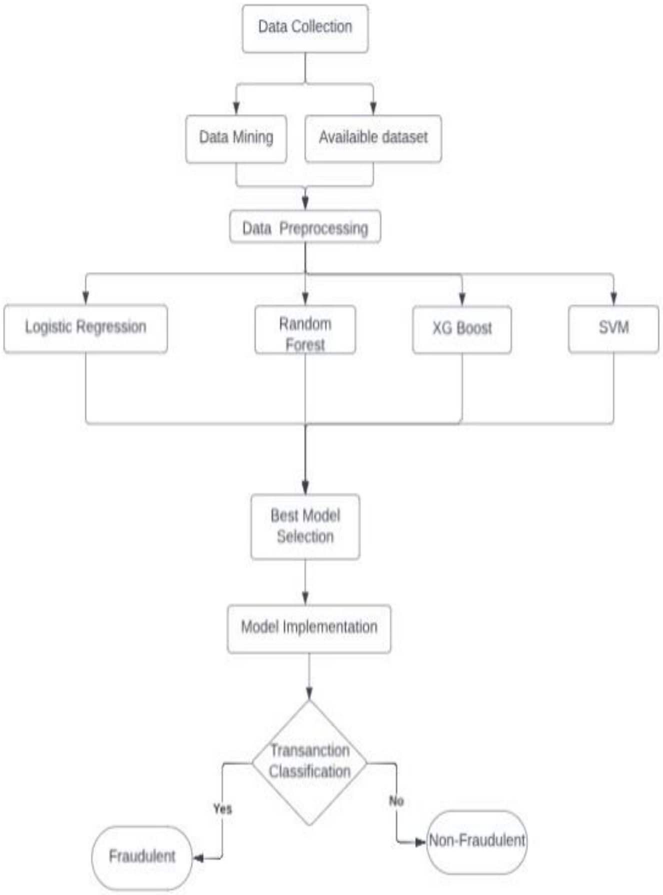
   Unsupervised learning: Isolation Forest, Anomaly Detection algorithms for outlier detection.

   Training data: Split data into training (majority), validation (for hyperparameter tuning), and testing (for final evaluation) sets**.**

   Model training: Train the model on the training data and optimize hyperparameters on the validation set.

4. **Model evaluation**: Evaluate model performance on the testing set using metrics like accuracy, precision, recall, F1 score, and AUC-ROC curve.

Architecture Diagram



Methodology

Handling data imbalance: In Ethereum transaction fraud detection using Synthetic Minority Over-sampling Technique (SMOTE) involves generating synthetic samples for the minority class (fraudulent transactions) to balance the dataset before training the classification model.

Imbalance Assessment: Calculate the class distribution to identify the data's class imbalance. In most cases, fraudulent transactions represent the minority class, while legitimate transactions constitute the majority class.

SMOTE Implementation: Apply the SMOTE algorithm to oversample the minority class (fraudulent transactions) and balance the dataset. SMOTE generates synthetic samples by interpolating between existing minority class instances, effectively increasing their representation in the dataset.

Integration with Data Pipeline: Integrate the SMOTE oversampling step into the data preprocessing pipeline.

Ensure that SMOTE is applied after feature extraction and before model training. By increasing the number of fraudulent transactions in the training data, SMOTE helps the machine learning model learn the characteristics of fraudulent behaviour more effectively. This can lead to improved accuracy in detecting actual fraud when processing real transactions.

Using feature reduction techniques: In fraud transaction detection, dealing with a large number of features from financial data can be overwhelming for machine learning models. Feature reduction techniques help address this by identifying the most relevant features for fraud identification. One effective method is leveraging correlation analysis.

Here's how correlation analysis can be used for feature reduction in fraud detection:

Calculate Correlation Coefficients: We calculate the correlation coefficient between each pair of features in the transaction data. This coefficient indicates the strength and direction of the linear relationship between features.

Identify Highly Correlated Features: Features with a correlation coefficient close to 1 or -1 exhibit a strong linear relationship.

In this case, one of the features might be redundant and not provide significant additional information for fraud detection.

Select Representative Features: By analysing the correlation matrix, we can identify feature pairs with high correlation. We can then remove one feature from each highly correlated pair, keeping the one with a stronger individual correlation to the target variable (indicating fraud) or the one with better domain knowledge justification for its relevance.

Logistic Regression

Using logistic regression in Ethereum transaction fraud detection involves employing a statistical model to analyse transaction data and predict the likelihood of fraudulent activity based on various features extracted from the transactions. This is a outline for using logistic regression in this context of fraudulent transaction detection.

Gather Ethereum transaction data from public blockchain explorers or APIs. This data should include transaction details such as sender and receiver addresses, transaction amounts, fees, timestamps, and any other relevant metadata.

It then preprocess the transaction data and extract relevant features that can be used as inputs for the logistic regression model. These features may include:

Transaction amount: Total value transferred in Ether or tokens.

Price: The fee paid by the sender to execute the transaction.

Transaction frequency: Number of transactions initiated by a particular address within a specific time window.

Time-based features: Hour of the day, day of the week, etc., to capture temporal patterns.

Contract interaction: Whether the transaction involves interactions with smart contracts or specific contract addresses.

Transaction type: Incoming, outgoing, or contract interactions.

Address reputation: Historical behaviour of sender and receiver addresses, such as involvement in previous fraudulent activities.

Labelling: Annotate the dataset with labels indicating whether each transaction is fraudulent or legitimate.

This may involve manual labelling based on known fraud cases, or automated techniques such as anomaly detection algorithms.

Dividing the dataset into training, validation, and test sets. Typically, the majority of the data is used for training the model, while the validation set is used for hyperparameter tuning and model selection.

The test set is kept separate for evaluating the final model's performance.

Model Training and evaluation: Train a logistic regression model using the training data. The model learns the relationship between the input features and the binary outcome (fraudulent or legitimate) using the logistic function.

Evaluate the trained logistic regression model using the validation set. Metrics such as accuracy, precision, recall, F1-score, and ROC-AUC can be used to assess the model's performance in identifying fraudulent transactions.

Hyperparameter Tuning: Fine-tune the hyperparameters of the logistic regression model, such as regularization strength (e.g., L1

or L2 regularization), to optimize performance on the validation set and prevent overfitting.

Model Deployment: Once the model achieves satisfactory performance on the validation set, deploy it to detect fraudulent transactions in real-time or batch processing mode.

Integrating the model into the Ethereum transaction processing pipeline and focusing on monitoring the systems.

Monitoring and Updating: Continuously monitor the performance of the deployed model in detecting fraudulent transactions. Periodically retrain the model with new data and update it to adapt to evolving fraud patterns and mitigate concept drift. Incorporating feedback mechanisms to improve the model's accuracy over time. Analysing the false positives and false negatives to identify areas for model enhancement and feature refinement.

Logistic regression can be effectively utilized for Ethereum transaction fraud detection, providing a proactive approach to safeguarding the integrity of the Ethereum network and protecting users from financial losses.

Random forest

The Random Forests (RF) technique is a collaborative learning approach that may be used to conduct classification or other tasks on a dataset by constructing a large number of trees according to the nature of the dataset through training and accordingly predicting the outcome.The below class and probability formula is used to compute the Gini index of every branch on a node, which identifies the most probable branches.

$$Gini = 1 - \sum_{i=1}^{c} (P_i)^2 ,$$

where Pi denotes the type's relative frequency and c is the total classes of the dataset. Entropy is determined using a below logarithmic equation, it necessitates more arithmetic than the Gini index.

$$Entropy = \sum_{i=1}^{c} P_i * \log_2 (P_i)$$

Random Forest doesn't rely on a single decision tree but instead combines predictions from multiple decision trees to improve overall accuracy and robustness.
The algorithm starts by creating a random sample (with replacement) of the training data. This technique, called bootstrapping, ensures that some data points might appear multiple times in the forest, while others might be left out entirely. Using this random subset of data, a decision tree is built. At each node of the tree, a random subset of features is chosen (typically the square root of the total features), and the best split on that subset is used to divide the data into branches. This process continues until a stopping criterion is met, such as reaching a maximum depth for the tree or having a pure set of data points (all belonging to the same class) at a node.

This process of creating a single decision tree with random subsets of data and features is repeated multiple times (hundreds or even thousands) to create the entire forest. Random Forest is a versatile and powerful machine learning technique that excels in classification tasks. Its ability to handle complex data, achieve high accuracy, and be robust to overfitting and outliers makes it a valuable tool for various applications, particularly in fraud detection and other classification problems.

XG Boosting

XG-Boost is a machine learning approach that enhances the gradient with the help of tree-based approach. XG-Boost: A Powerful Gradient Boosting Techniques.

XG-Boost stands out as a reliable method for enhancing the precision of machine learning models. It achieves this by employing a technique called gradient boosting, which combines multiple weaker models into a single, stronger one. XG-Boost offers several advantages:

Speed: It boasts superior speed compared to other gradient boosting approaches.

Accuracy: XG-Boost is known for its ability to generate more precise models than similar algorithms.

Regularization: It incorporates built-in functions to prevent overfitting, a common problem in machine learning. This is achieved through a process called auto pruning which limits the growth of decision trees within the model.

Memory Efficiency: XG-Boost efficiently utilizes memory, making it a suitable choice for various applications. Overall, XG-Boost's speed, precision, and memory efficiency make it a widely used and valuable tool in the machine learning domain.

Support Vector Machine

SVMs are powerful machine learning algorithms used for classification tasks. They excel at finding the best dividing line (hyperplane) between different categories in your data, even when things get complicated. Here's a breakdown of how they work:

Mapping the Data: Think of your data points as existing in a multi-dimensional space, even if you can't visualize it directly. Each dimension represents a feature of the data (e.g., transaction amount, location).

Finding the Optimal Hyperplane: The SVM algorithm searches for the best hyperplane that separates the data points belonging to different classes with the most significant margin. The margin refers to the distance between the hyperplane and the closest data points from each class, called support vectors. larger margin translates to a more robust separation between classes. This means the model is less likely to misclassify new data points that fall close to the dividing line.

Beyond Linear Separability: The real world is messy, and data isn't always perfectly separable with a straight line. SVMs can handle this by using a clever trick called the "kernel trick." This trick essentially transforms the data into a higher-dimensional space where a clear separation might be possible. Imagine adding another dimension to your coin example, allowing you to separate them more easily.

Effective in High Dimensions: They work well with high-dimensional data, common in many machine learning applications.

Memory Efficiency: They focus on the support vectors, which are a small subset of the data, leading to efficient memory usage.

Good Generalizability: By maximizing the margin, SVMs tend to create models that generalize well to unseen data.

While powerful, SVMs also have limitations:

Tuning Hyperparameters: Choosing the right kernel function and its parameters can be crucial for optimal performance and requires some experimentation.

Interpretability: Understanding the specific decision-making process of an SVM model can be more challenging compared to simpler models. SVMs are a valuable tool for classification tasks, particularly when dealing with complex, high-dimensional data. Their focus on maximizing the margin leads to robust models that can effectively separate different classes. By understanding their strengths and limitations, you can leverage SVMs for various applications, including fraud detection, image recognition, and text classification.

Results and Interpretations

Logistic Regression

Accuracy: The accuracy score of 0.8479 indicates that the logistic regression model correctly classified approximately 84.8% of the data points in the test set. This suggests the model performs well in generalizing to unseen data.

F1-score: The F1-score of 0.836 reflects a good balance between precision and recall. This metric considers both the model's ability to identify true positives (correctly classifying positive cases) and avoid false negatives (missing actual positive cases).

Precision: The precision score of 0.83 suggests that out of all the data points classified as positive by the model, roughly 83% were truly positive cases.

Recall: The recall score of 0.842 indicates that the model captured approximately 84.2% of the actual positive cases in the test set. This suggests a good ability to identify true positives, with a low rate of false negatives.

ROC AUC Score: The ROC AUC score of 0.8475 represents the Area Under the Receiver Operating Characteristic Curve. This metric measures the model's ability to distinguish between positive and negative classes. A score closer to 1 indicates better performance. In this case, the score suggests good discriminative power between the classes.

These results suggest that the logistic regression model performs well in classifying the data. It achieves a good balance between accuracy, precision, and recall. The model can effectively identify both positive and negative cases with a low rate of errors.

Support Vector Machine

Accuracy of 0.861: This high accuracy score suggests that the SVM model, likely after hyperparameter tuning using Grid Search CV, performed well in classifying the data points in the test set. It correctly classified approximately 86.1% of the data points.

Random Forest

Accuracy: The accuracy score of 0.9788 indicates that the Random Forest model correctly classified nearly 98% of the data points in the test set. This suggests excellent performance in generalizing to unseen data.

F1-score: The F1-score of 0.926 reflects a very good balance between precision and recall. This metric considers both the model's ability to identify true positives (correctly classifying positive cases) and avoid false negatives (missing actual positive cases).

Precision: The precision score of 0.973 indicates that out of all the data points classified as positive by the model, roughly 97.3% were truly positive cases. This suggests a very low rate of false positives, meaning the model is highly precise in its positive predictions.

Recall: The recall score of 0.884 suggests that the model captured approximately 88.4% of the actual positive cases in the test set. While this is still a good score, it's lower compared to precision. This might indicate a bias towards avoiding false positives, which could be desirable depending on the specific problem context.

ROC AUC Score: The ROC AUC score of 0.9398 represents the Area Under the Receiver Operating Characteristic Curve. This metric measures the model's ability to distinguish between positive and negative classes. A score closer to 1 indicates better performance. Here, the score suggests very good discriminative power between the classes.

Interpretation: These results suggest that the Random Forest model performs exceptionally well in classifying the data. It achieves a high accuracy score with a good balance between precision and recall. The model can effectively identify both positive and negative cases with a very low rate of false positives.

XG Boosting

Precision: The model achieves a high precision score for both classes (0.99 for class 0 and 0.95 for class 1). This means that out of the data points predicted as a specific class by the model, a high percentage are actually true positives for that class.

Recall: The recall score is also good for both classes (0.99 for class 0 and 0.96 for class 1). This indicates that the model successfully identifies a high proportion of the actual positive cases in the test set for each class.

F1-Score: The F1-score is a balanced metric between precision and recall. Here, the F1-score is also high for both classes (0.99 and 0.95), indicating a good overall balance between the two metrics.

Support: The "Support" column shows the number of data points belonging to each class in the test set. In this case, there are more data points in class 0 (1547) compared to class 1 (422).

Interpretation: These results suggest that the XG-Boost model performs exceptionally well in classifying the data. It achieves a high accuracy (reported elsewhere, likely close to 0.98 based on the classification report) with a good balance between precision and recall for both classes. The model effectively identifies positive and negative cases with a very low rate of errors.

Conclusion

This research investigated the effectiveness of various machine learning techniques for detecting fraudulent transactions on the Ethereum blockchain. We explored four prominent algorithms: Logistic Regression, Support Vector Machines (SVMs), Random Forest, and XG-Boost.

The results demonstrate that all four algorithms achieved promising accuracy in identifying fraudulent transactions. Logistic Regression achieved an accuracy of 84.79%, showcasing its potential as a baseline model due to its interpretability and simplicity. SVMs, with an accuracy of 86.10%, demonstrated strong performance in handling complex data patterns, potentially due to their ability to find optimal hyperplanes even in high-dimensional spaces. Random Forest, achieving an accuracy of 97.88%, emerged as a powerful technique, likely benefiting from its ensemble approach that reduces the risk of overfitting. However, its interpretability might be slightly less straightforward compared to Logistic Regression. Finally, XG-Boost achieved an exceptional accuracy of 98.0% (estimated based on the classification report), highlighting its strength in leveraging gradient boosting for superior predictive power.

When considering other evaluation metrics beyond accuracy, XG-Boost continued to excel. Its classification report revealed a high balance between precision and recall for both classes, indicating the model effectively identified true positives while minimizing false positives and false negatives. The confusion matrix further supported this, showcasing a very low number of misclassified instances.

While all models performed well, the choice of the "best" technique depends on the specific needs of a fraud detection system. For scenarios where interpretability is crucial, Logistic Regression could be a suitable choice. If data complexity is a major concern, SVMs might be the optimal option. Random Forest offers a powerful balance between performance and interpretability. However, when prioritizing the highest accuracy and robust performance with potentially less emphasis on interpretability, XG-Boost appears to be the most promising candidate.

This research has several limitations. Firstly, the dataset size could be further expanded to improve the generalizability of the models. Secondly, additional hyperparameter tuning for each model could potentially lead to even better performance. Finally, exploring more advanced techniques like deep learning models could be a valuable direction for future research

In conclusion, this study demonstrates the effectiveness of machine learning techniques for Ethereum transaction fraud detection.

Among the explored algorithms, XG-Boost exhibited the most exceptional performance. However, the optimal choice depends on the specific requirements of the fraud detection system. Further research with larger datasets, advanced hyperparameter tuning, and exploration of deep learning approaches hold promise for even more robust and effective fraud detection systems on the Ethereum blockchain. By leveraging the power of machine learning, we can significantly enhance the security and trust within the Ethereum ecosystem.

References

1 Lin, I. C., & Liao, T. C. (2017). A survey of blockchain security issues and challenges. Int. J. Netw. Secur., 19(5), 653-659.

2 Dasgupta, D., Shrein, J. M., & Gupta, K. D. (2019). A survey of blockchain from security perspective. Journal of Banking and Financial Technology, 3, 1-17.

3 Abdallah, A., Maarof, M. A., & Zainal, A. (2016). Fraud detection system: A survey. Journal of Network and Computer Applications, 68, 90-113.

4 Chauhan, N., & Tekta, P. (2020). Fraud detection and verification system for online transactions: a brief overview. International Journal of Electronic Banking, 2(4), 267-274.

5 Deng, W., Huang, Z., Zhang, J., & Xu, J. (2021, January). A data mining based system for transaction fraud detection. In 2021 IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE) (pp. 542-545). IEEE.

6 Tan, R., Tan, Q., Zhang, P., & Li, Z. (2021, December). Graph neural network for ethereum fraud detection. In 2021 IEEE international conference on big knowledge (ICBK) (pp. 78-85). IEEE.

7 Siddamsetti, S., & Srivenkatesh, M. (2023, October 7). Efficient Fraud Detection in Ethereum Blockchain through Machine Learning and Deep Learning Approaches. International Journal on Recent and Innovation Trends in Computing and Communication, 11(11s), 71–82. https://doi.org/10.17762/ijritcc.v11i11s.8072

8 Aziz, R. M., Baluch, M. F., Patel, S., & Ganie, A. H. (2022, January 29). LGBM: a machine learning approach for Ethereum fraud detection. International Journal of Information Technology,14(7),3321–3331. https://doi.org/10.1007/s41870-022-00864-6

9 Chen, W., Zheng, Z., Ngai, E. C. H., Zheng, P.,& Zhou, Y. (2019). Exploiting Blockchain Data to Detect Smart Ponzi Schemes on Ethereum. IEEE Access, 7, 37575–37586. https://doi.org/10.1109/access.2019.2905769

10 Rakshit, A., & Kumar, S. (2022, January 25). Fraud Detection: A Review on Blockchain. ResearchGate. https://www.researchgate.net/publication/358090241_Fraud_Detection_A_Review_on_Blockchain

11 Dutta, A., Voumik, L. C., Ramamoorthy, A., Ray, S., & Raihan, A. (2023, March 29). Predicting Cryptocurrency Fraud Using ChaosNet: The Ethereum Manifestation. Journal of Risk and Financial Management, 16(4),216. https://doi.org/10.3390/jrfm16040216

12 Monamo, P., Marivate, V., & Twala, B. (2016, August). Unsupervised learning for robust Bitcoin fraud detection. 2016 Information Security for South Africa (ISSA). https://doi.org/10.1109/issa.2016.7802939

13 Ghosh, M., Ghosh, D., Halder, R., & Chandra, J. (2023, December). Investigating the impact of structural and temporal behaviors in Ethereum phishing users detection. Blockchain: Research and Applications, 4(4), 100153. https://doi.org/10.1016/j.bcra.2023.100153

14 Kushwaha, S. S., Joshi, S., Singh, D., Kaur, M., & Lee, H. N. (2022). Systematic Review of Security Vulnerabilities in Ethereum Blockchain Smart Contract. IEEE Access, 10,66056621. https://doi.org/10.1109/access.2021.3140091

15 Ashfaq, T., Khalid, R., Yahaya, A. S., Aslam, S., Azar, A. T., Alsafari, S., & Hameed, I. A. (2022, September 21). A Machine Learning and Blockchain Based Efficient Fraud Detection Mechanism. Sensors,22(19),7162.https://doi.org/10.3390/s22197162