[1]Teng Yuan

Combating Fraud in Decentralized Finance: A Comprehensive Feature Engineering Scheme for Machine Learning-Based Detection

Abstract: - This paper conducts a systematic and in-depth study on the increasingly severe fraudulent transaction problem in the field of Decentralized Finance (DeFi). First, the DeFi ecosystem is carefully examined, focusing on the analysis of Ethereum's architecture, key technical elements, and the layered architecture of DeFi applications. On this basis, core issues such as smart contract security and token system design are discussed, and the technical challenges currently faced by DeFi fraudulent transaction detection are analyzed. To address the limitations of existing research in effectively modeling the complex dynamic associations between transaction entities and the evolution of fraud patterns over time, this paper innovatively proposes a DeFi fraudulent address detection method based on machine learning.

This method focuses on multiple aspects such as transaction behavior, capital flow, and account attributes, designing a comprehensive feature engineering scheme. It extracts 35 key features closely related to fraud detection and further optimizes them to 27 features through wrapped feature selection, constructing a complete and concise feature set.

Different from existing research that mainly relies on the analysis of ordinary transactions and ERC20 token transactions, this paper additionally introduces internal transaction features. Although internal transactions are not directly recorded on the blockchain, they contain rich user behavior information. By capturing and utilizing internal transaction information, this paper further improves the performance of fraud detection models.

Multiple machine learning models are employed and their performance on this task is analyzed and compared. The paper selects four models: K-Nearest Neighbor, Random Forest, XGBoost, and LightGBM, and trains and tests them using the feature set from existing research and the feature set constructed in this paper respectively. Experimental results show that under most models, the performance using the feature set of this paper is superior to using the existing feature set, verifying the effectiveness of the feature engineering scheme proposed in this paper. Among them, the LightGBM model achieves the best overall performance.

This paper conducts fruitful explorations in the field of DeFi fraudulent transaction detection. The proposed theoretical methods, technical models, and practical systems provide key technical support for the security governance of the DeFi ecosystem.

Keywords: Reinforcement Learning, Tennis, Optimization, Tactical Decision, Ranking, Genetic Algorithm

1. Introduction

1.1 Background

In 2013, Vitalik Buterin proposed the Ethereum platform, which marked the entry of a new phase in block chain technology. Ethereum not only retains the function of digital currency transactions, but also introduces smart contract technology based on virtual machines, which supports Turing completeness, making it possible to run

[1] [1]* School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing, 210000, Jiangsu, China.Email:yuant6666@126.com

complex programs on the blockchain and greatly expanding the scope of blockchain applications.

In the field of Decentralized Finance (DeFi), with its rapid growth, its security issues have become increasingly prominent[7]. Although smart contracts guarantee the immutability of data and a transparent process, they may also have security vulnerabilities that could be exploited by attackers to steal funds. DeFi and related security issues have become the focus and difficulty of block chain research and practical application, and its security is a key factor to promote the development of block chain technology.

1.2 Research Actuality

1.2.1 Transaction Security Detection

The logic of DeFi applications is so complex that traditional smart contract vulnerability detection tools struggle to cope with it. Attacks involving smart contract combinations or logical vulnerabilities, in particular, are difficult to detect through conventional techniques. By analyzing transaction data related to DeFi, some scholars extract key information and features to analyze potential security risks[23]. This paragraph mainly reviews the research progress of such technology.

Siwei Wu[24] et al. proposed a method of using Ethereum transaction data to build digital asset cash flow tree (CFT) for advanced semantic reconstruction to detect attacks such as price manipulation. By constructing CFT mapping asset transfer, the potential attack behavior is further identified by pattern matching. The method focuses on the detection of price manipulation attacks and has validated its effectiveness in hundreds of millions of transaction data.

Liyi Zhou[25] et al. developed automated tools for constructing arbitrage behaviors and quantifying transactions through Markov decision processes to identify possible attack behaviors. While it is debatable whether the arbitrage constitutes an attack, its impact on the Ethereum community is clear. They use circular search and heuristic pruning to optimize the search process and enhance the detection of price manipulation.

Dabao Wang[26] et al. studied the detection method of lightning loan attack in DeFi applications. They screen transactions related to flash lending and analyze transaction intent and motivation by defining transaction patterns and DeFi application primitives (such as exchange, lending, etc.), improving the recognition rate of such attacks.

Yixin Cao[28] et al. developed Flashot, a tool that analyzes the flow of assets in DeFi applications through smart contract interaction. By defining basic trading elements and operations, Flashot visualizes the trading process, helping to understand attack steps in detail and tracking asset flows.

Wang[29] et al. analyze function calls at the Ethereum virtual machine level to identify possible predictor vulnerabilities, while adopting a rule-based approach to monitor transactions and flag suspicious activity.

The existing methods are still limited in detection scope and timeliness, and it is difficult to cover the diversity and dynamics of DeFi applications. As a result, the solution strategy is usually only for a few known security problems, and it is difficult to deal with the emerging complex security challenges.

1.3 The Main Innovation of This Paper

A DeFi scam address detection method based on machine learning is proposed. This paper designs a comprehensive feature engineering scheme from the perspectives of transaction behavior, fund flow, account attributes, etc., extracts 35 key features that are closely related to fraud detection, and further optimizes them into 27 features through encapsulated feature selection to construct a comprehensive and simplified feature set.

Internal transaction features are introduced. Unlike the existing studies, which mainly rely on the analysis of ordinary transactions and ERC20 token transactions, this paper introduces an additional internal transaction feature. Although internal transactions are not directly recorded on the block chain, they contain a wealth of information about user behavior. This paper further improves the performance of fraud detection model by capturing and utilizing internal transaction information.

A variety of machine learning models are used and their performance on this task is compared and analyzed. Four models, K proximity, random forest, XG Boost and Light GBM, are selected for training and testing using the feature sets previously studied and the feature sets constructed in this paper. The experimental results show that under most models, the performance of the Venter collection is better than that of the existing feature set, which verifies the effectiveness of the feature engineering scheme in this paper. The Light GBM model has achieved the best comprehensive performance.

2. Background Knowledge Introduction

2.1 Ethereum

Blockchain technology is a distributed ledger technology. Its feature is that once the data is recorded, it cannot be changed, thus ensuring the security and immutability of the information. This technique is very effective in peer-to-peer networks in terms of ensuring the accuracy of transaction information and the tracking of assets. The security of the blockchain is due to its encryption algorithm, and transactions within each block need to be confirmed by a majority of nodes in the network. Since its inception in 2015, the Ethereum platform has rapidly developed into the world’ s second largest cryptocurrency system after Bitcoin. This article aims to delve into Ethereum’ s key features, architecture, smart contract capabilities, and its place in the application of blockchain technology.

1.3.1 2.1.1 Ethereum Account

Ethereum accounts can be divided into two categories, user accounts and contract accounts, depending on their features and properties[33]. User accounts are controlled by private keys that require account holders to pass key verification before they can participate in Ethereum network activities, such as sending transactions, deploying contracts, etc. The contract account corresponds to the smart contract one by one, and its line is determined by the contract code, and it can autonomously perform operations such as transfer and call other contract functions. Both the user account and the contract account contain basic information such as the number of transactions of the account, the ether balance, and the 256-bit hash value pointing to the location where the data is stored. In addition, the contract account also stores an additional smart contract bytecode hash deployed at that address, which is used to verify contract authenticity when the transaction is called.

1.3.2 2.1.2 Ethereum Transactions

Transactions on Ethereum are operational instructions initiated by external accounts and are the primary way users interact with the Ethereum network. Transactions can carry binary data and ether, which are used to modify or update the state in the Ethereum network. According to the different originators and recipients of transactions, transactions can be divided into external transactions and internal transactions. External transactions are initiated by an external account, the target address is also an external account, and such transactions are packaged directly into the block and become part of the blockchain. Internal transactions are triggered by external transactions, resulting from the execution of smart contracts, and are not recorded directly on the blockchain. In the transaction process of smart contract participation, the contract can issue events to record the key state of the contract operation, which is convenient for developers to track and analyze the contract behavior.

1.3.3 2.1.3 Ethereum Virtual Machine

2.2 ERC-20 tokens

In the blockchain world, digital assets, also known as digital tokens, are a type of cryptocurrency that utilizes smart contracts. The variables in the smart contract are used to record relevant information about the holder of the digital asset, including the address, the amount of ownership, etc[52-53]. The standard defines a uniform set of interfaces and specifications that enable different tokens to be compatible and interoperate within the Ethereum ecosystem[54]. The two most critical functions in the ERC20 standard are transfer() and transferFrom(), and a Transfer event. The transfer() function is used to transfer digital assets from the current account to another specified address. When this function is called, the token balance in the sender's account will decrease and the token balance in the receiver’ s account will increase accordingly. The transferFrom() function works similarly to transfer(), but it allows the token holder to authorize another account to transfer the token on its behalf. This mechanism provides more flexibility for the smart contract interaction of tokens. When the Transfer operation is completed, the smart contract triggers a transfer event, which is used to notify other participants in the blockchain network about the token transfer, such as the sender address, the receiver address, and the number of transferred tokens. This event mechanism makes the token transfer process more transparent and traceable.

In addition to the above two core functions, the ERC20 standard defines other functions that must be implemented, such as balanceOf() to query the token balance of a given account, approve() to authorize another account to transfer tokens on behalf of another account, allowance() to query the authorization amount, and so on [55]. These standardized interfaces provide a convenient way of token integration for various DApps on Ethereum, which greatly promotes the prosperity and development of the Ethereum ecosystem.

3. DeFi Fraud Address Detection Method Based on Machine Learning

3.1 Introduction

At present, the mainstream fraud address detection method is based on supervised learning classification model. By analyzing the Ethereum address, these models extract a series of indicators that reflect the behavior features of the address, such as transaction frequency, transaction amount, transaction object, etc. These features are then

fed into a trained classifier to determine whether the address is a scam address. Although this method has achieved some results, its performance is still limited by the ability of feature representation. Existing studies rely on analysis of both ordinary transactions and ERC-20 token transactions when constructing feature vectors. Insider trading refers to currency transfers triggered during the execution of smart contracts. Internal transactions are not recorded directly on the blockchain, but contain a wealth of information about user behavior. If we can capture and use this information effectively, we can further improve the performance of fraud detection model.

This paper introduces the internal transaction feature, and reconstructs and optimizes the existing feature system. Firstly, a large amount of real transaction data is collected to achieve comprehensive feature extraction and screening. Through further feature engineering and model tuning, an optimal feature combination for DeFi fraud detection is finally obtained, and excellent detection results are obtained on the data set.

3.2 Model Introduction

This paper proposes an Ethereum scam address detection model based on multi-factor features of transactions. The model mainly consists of three modules: data acquisition, feature extraction and classification detection.

The data acquisition module is responsible for collecting raw data for model training and testing. On the one hand, it is necessary to collect confirmed fraud addresses as positive samples; On the other hand, a certain number of normal addresses need to be collected as negative samples. The two types of address data together constitute the training set and the test set.

Feature extraction module: After the completion of the original data collection, it is necessary to extract effective features from the original transaction data. By analyzing common Ethereum fraud methods, this paper designs a comprehensive feature engineering scheme from the perspectives of transaction behavior, fund flow, account attributes and so on. After extracting all kinds of initial features, it is necessary to further screen features, delete redundant and invalid features, reduce the feature dimension, and improve the generalization performance of the model.

Based on the feature matrix obtained by feature engineering, the classification detection module constructs a machine learning classification model to realize fraud judgment of new addresses. Through comparative experiments, this paper adopts integrated learning algorithms, such as XGBoost, LightGBM, etc. This kind of algorithm not only has excellent classification performance, but also can adapt well to problems such as sparse features and unbalanced samples, so it is very suitable for the application scenario of this project.

Through the collaboration of the above three modules, this paper hopes to build a set of superior performance of Ethereum scam address detection model. Compared with the existing methods, the model adopts a more comprehensive feature engineering scheme and can depict the behavior pattern of fraudulent addresses from multiple angles. At the same time, the model makes full use of the domain knowledge of DeFi scene, and carries out targeted optimization on the features and models, which is expected to achieve more accurate detection effect.

3.2.1 Data Acquisition Module

In order to construct a dataset for DeFi scam user identification, this paper makes use of multiple data sources. The scam address data used in this article comes mainly from two sources: the underlying data set provided by BERT4ETH, and the detailed data scraped from Etherscan, the Ethereum blockchain browser, through a self-developed crawler.

During the crawler implementation, the template URL (the entry address where the crawler runs) is first defined. The list of urls to be crawled is obtained by de-processing the template URL and the crawled webpage. In order to improve the efficiency of web page fetching and prevent data download failure caused by network fluctuation, this paper sets a retry mechanism.

In order to avoid the impact of the anti-crawling mechanism, when sending a request to Etherscan, the crawler simulated the browser header information, disguised as an ordinary user to visit, and successfully obtained the webpage content corresponding to the URL. After analytic processing, the required structured data is finally obtained. In order to facilitate subsequent analysis, this paper converts and exports the obtained web data. By saving the data in csv format.

3.2.2 Feature Extraction Module

Firstly, this paper focuses on the transaction behavior features of addresses. This kind of feature includes the transaction frequency, transaction amount and transaction time distribution of the address. The transaction frequency reflects the activity of the address, while the transaction amount reflects the economic strength and willingness of the address to trade. The distribution features of transaction time can help this paper to observe the transaction habits and rules of addresses. Through comprehensive analysis of these transaction behavior features, this paper can better understand the role and behavior pattern of address in the Ethereum network.

Secondly, this paper also deeply analyzes the features of the fund flow of the address. By calculating the inflow and outflow ratio of funds at an address, this paper can judge whether the source and destination of funds at the address are balanced, and then infer whether there may be abnormal transactions. In addition, the paper also counts the number of accounts where the address funds come from and go, as well as the distribution of funds in different tokens held by the address. These features can help this paper to fully understand the fund interaction of the address, and provide important clues for abnormal transaction detection.

Table 3.1 Extracted Features

Feature	Description
erc20_tx_count	The total number of ERC-20 token transfer transactions initiated by this address
erc20_rx_count	The total number of ERC-20 token transfer transactions received at this address
erc20_rx_address_count	The number of different addresses to which ERC-20 tokens are transferred
erc20_first_last_tx_time_diff	The time gap between the first and last ERC-20 token transfer

	transaction of the address
erc20_tx_min_amount	Minimum amount of ERC-20 token transfer transaction initiated by this address
erc20_tx_max_amount	Maximum amount of ERC-20 token transfer transaction initiated by this address
erc20_tx_avg_amount	Average amount of ERC-20 token transfer transactions initiated by the address
internal_total_count	The total number of internal transactions in which the address is involved
internal_rx_address_count	The number of different addresses that initiate internal transactions to the address
internal_tx_address_count	The number of internal transactions initiated by the address to different addresses
internal_rx_max_amount	The maximum amount of internal transactions received by the address
internal_rx_avg_amount	The average amount of internal transactions received by the address
normal_tx_count	The number of normal transactions initiated by the address
normal_rx_count	The number of ordinary transactions received by the address
normal_rx_address_count	The number of different addresses to which ordinary transactions are transferred to the address
normal_rx_total_amount	The total amount of ordinary transactions received by the address
out_in_tx_ratio	The ratio of the number of transactions transferred out of the address to the number of transactions transferred into the address
high_risk_erc20_transfer_count	The number of times the address transfers high-risk ERC-20 tokens
high_risk_erc20_transfer_amount	The address transfers the total amount of high-risk ERC-20 tokens
blacklisted_address_interaction_count	The number of times the address interacts with the blacklisted address
blacklisted_address_interaction_amount	The amount that the address interacts with the blacklisted address
flashloan_count	The number of flash loans the address participated in
flashloan_amount	The total amount of flash loans that the address participated in
balance_total_eth_ratio	The proportion of the address balance to the total number of ether received
normal_rx_avg_amount	The average amount of ordinary transactions received by the address
out_in_tx_ratio	The ratio of the number of transactions initiated by the address to the number of transactions received
uniswap_total_tx_count	The total number of transactions of the address on Uniswap
ethereum_total_tx_count	The total number of transactions on Ethereum for that address
uniswap_mint_count	The total number of mint events triggered by the address on Uniswap

uniswap_swap_count	The total number of swap events triggered by the address on Uniswap
uniswap_swap_to_count	The total number of events that the address is on Uniswap as a swap source
uniswap_swap_from_count	The total number of events that the address is on Uniswap as a swap source
uniswap_swap_from_count	Total number of swap-from events on Uniswap

In addition to trading behavior and money flow characteristics, this paper also focuses on account attribute characteristics of addresses. The account creation time reflects the history of the address, while the account balance reflects the wealth of the address. This paper also counts the number of contracts in which an address participates and the number of ERC-20 tokens it holds. These account attributes can help this paper to describe the characteristics of addresses more comprehensively and provide more valuable information for fraud detection.

Based on the above feature extraction strategies, this paper finally extracts 35 key features that are closely related to fraud detection, as shown in Table 3.1. Through wrapper feature selection, we further optimize the feature set and eliminate those redundant features that contribute little to fraud detection, so as to obtain a compact and effective feature set. These selected features will provide strong support for subsequent fraud detection models and help to improve the accuracy and reliability of detection.

3.3 Related Machine Learning Algorithms

3.4 Experiments on DeFi Fraud Address Detection

3.4.1 Data Set

The fraud account data comes from the basic data set provided by BERT4ETH, which contains all kinds of illegal address information based on Ethereum. By screening the account address, 5673 fraud address samples are obtained. The normal account data came from the Ethereum main network. By randomly selecting part of the transaction accounts within the block of 12500000 to 14500000, and eliminating the known fraud addresses, 5798 normal address samples were finally obtained.

In order to obtain the transaction data of these addresses, this paper uses the API interface provided by Ethereum browser Etherscan to grab the ordinary transaction records, internal transactions and ERC-20 token transaction records of each address, and extracts the required features from them. After the data preprocessing was completed, a data set containing 11471 address feature data was finally obtained. Considering the requirements of training data and test data for machine learning models, this paper uses the ratio of 7:3 to divide the dataset into training set and test set. This paper constructs a comprehensive and reliable DeFi fraud user dataset, which lays a data foundation for subsequent machine learning model training.

3.4.2 Feature Selection

In order to select the optimal feature subset from the original feature set, this paper uses the feature selection technology to optimize the features, in order to reduce the feature dimension and improve the recognition

performance of the fraud detection model.

In this study, the wrapper feature selection method based on XGBoost classifier is used in this paper, and the feature set is optimized by backward search strategy. Specifically, the feature selection process starts from the complete set containing all candidate features, and tries to eliminate unimportant features by evaluating the importance of the model for each feature in each iteration. In order to judge whether a feature should be eliminated, this paper retrains the evaluation model after eliminating the feature, and evaluates the performance on the test set. If the model performance does not decrease significantly after removing the feature, it is considered that the feature has little contribution to the model and can be safely removed. On the other hand, if removing the feature causes a significant decrease in model performance, the feature should be retained. Through this recursive feature compression process, a preferred feature subset is finally obtained in this paper, as shown in Table 3.2.

Table 3.2 Feature Sets for Further Screening

Feature	Description
erc20_tx_count	The total number of ERC-20 token transfer transactions initiated by this address
erc20_rx_count	The total number of ERC-20 token transfer transactions received at this address
erc20_rx_address_count	The number of different addresses to which ERC-20 tokens are transferred
erc20_first_last_tx_time_diff	The time gap between the first and last ERC-20 token transfer transaction of the address
erc20_tx_min_amount	Minimum amount of ERC-20 token transfer transaction initiated by this address
erc20_tx_max_amount	Maximum amount of ERC-20 token transfer transaction initiated by this address
erc20_tx_avg_amount	Average amount of ERC-20 token transfer transactions initiated by the address
internal_total_count	The total number of internal transactions in which the address is involved
internal_rx_address_count	The number of different addresses that initiate internal transactions to the address
internal_tx_address_count	The number of internal transactions initiated by the address to different addresses
internal_rx_max_amount	The maximum amount of internal transactions received by the address
internal_rx_avg_amount	The average amount of internal transactions received by the address
normal_tx_count	The number of normal transactions initiated by the address
normal_rx_count	The number of ordinary transactions received by the address

normal_rx_address_count	The number of different addresses to which ordinary transactions are transferred to the address
normal_rx_total_amount	The total amount of ordinary transactions received by the address
out_in_tx_ratio	The ratio of the number of transactions transferred out of the address to the number of transactions transferred into the address
high_risk_erc20_transfer_count	The number of times the address transfers high-risk ERC-20 tokens
high_risk_erc20_transfer_amount	The address transfers the total amount of high-risk ERC-20 tokens
blacklisted_address_interaction_count	The number of times the address interacts with the blacklisted address
blacklisted_address_interaction_amount	The amount that the address interacts with the blacklisted address
balance_total_eth_ratio	The proportion of the address balance to the total number of ether received
normal_rx_avg_amount	The average amount of ordinary transactions received by the address
out_in_tx_ratio	The ratio of the number of transactions initiated by the address to the number of transactions received
uniswap_total_tx_count	The total number of transactions of the address on Uniswap
ethereum_total_tx_count	The total number of transactions on Ethereum for that address
uniswap_mint_count	The total number of mint events triggered by the address on Uniswap
uniswap_swap_count	The total number of swap events triggered by the address on Uniswap

3.4.3 Evaluation Index

In order to objectively evaluate the performance of the constructed fraud address detection model, this paper uses several evaluation metrics, including Precision, Recall, and F1 value. These indicators describe the discriminative ability and generalization performance of the model from different dimensions.

The precision rate reflects how accurate the model is in identifying scam addresses. It is calculated as follows.

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \tag{3.1}$$

TP represents the number of real scam addresses that are correctly identified, and then represents the number of normal addresses that are wrongly determined to be scam addresses. The higher the accuracy rate, the more accurate the model is in discriminating fraudulent addresses and the lower the false alarm rate.

Recall measures the ability of the model in detecting all scam addresses. It is calculated as follows:

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \tag{3.2}$$

FN represents the number of scam addresses that were missed. The higher the recall rate, the more fraudulent addresses can be identified by the model and the lower the false negative rate.

The F1 score is the harmonic mean of precision and recall, taking into account both metrics. It is calculated as follows:

$$F1 = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3.3)$$

A higher F1 value indicates that the model achieves a better balance between precision and recall and has a stronger overall performance.

Through these evaluation indicators, the performance of the proposed fraud address detection model can be comprehensively and objectively evaluated, which provides a quantitative basis for the optimization and application of the model. At the same time, these metrics also provide a unified standard for performance comparison between different models, which helps to select the optimal detection scheme.

3.4.4 Experimental Result

Table 3.3 Table 3.3 Comparison of Experimental Results

Feature	Precision	Recall	F1 value
KNN-1	0.90837	0.86528	0.8863
KNN-2	0.92016	0.87476	0.89689
RF-1	0.91605	0.93169	0.9238
RF-2	0.9212	0.93169	0.92642
XGBoost-1	0.95769	0.94497	0.95129
XGBoost-2	0.95437	0.95256	0.95347
LightGBM-1	0.95585	0.94497	0.95038
LightGBM-2	0.95962	0.94687	0.95322

In the experiment, this paper selects four different models, including K-proximity algorithm, random forest, XGBoost and LightGBM. For each model, the feature set selected by reference [21] (feature set 1) and the feature set extracted in this paper (feature set 2) are used for training and testing, respectively. 1 Feature set 1 contains 42 transaction information features, while feature set 2 introduces internal transaction information and performs feature selection on all extracted features, resulting in 27 features.

The experimental results show that compared with the model using feature set 1, the model using this feature set has higher F1 score and Recall. It shows that the feature extraction method in this paper can improve the detection performance of Ethereum scam addresses. It is worth noting that, except random forest, all the other models achieve higher recall when using feature set 2, meaning that more scam addresses are successfully identified. Given that Ethereum scams can cause significant financial losses to victims, each step of model performance improvement is significant.

Among all the models, LightGBM algorithm has the best overall performance, the highest Precision and F1 value. This result shows that ensemble learning methods, such as random forest and XGBoost, have more advantages than traditional k-nearest neighbor algorithms in the task of Ethereum scam address detection.

In general, the feature extraction method based on blockchain transaction information proposed in this paper can effectively improve the detection performance of Ethereum scam addresses. By introducing internal transaction information and feature selection, this paper constructs a more comprehensive and simplified feature set.

Experimental results show that the feature set can achieve better comprehensive detection results under different models. It provides a new idea for DeFi fraud address detection and helps to further improve the blockchain security protection system.

References

[1] Zhang H Y, Zheng L, Cai L. Design and Analysis of Hierarchical Physical Layer Network Coding[J]. IEEE Transactions on Wireless Communications, 2017, 16(12): 7966-7981.

[2] Zhang Xianda, Bao Zheng. Communication Signal Processing [M]. Beijing: National Defense Industry Press, 2000: 30-50.

[3] Larimore M, Treichler J. Convergence behavior of the constant modulus algorithm[C]. IEEE International Conference on ICASSP,2017:13-16.

[1] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system[J]. Decentralized Business Review, 2008, 21260.

[2] Shao Qifeng, Jin Cheqing, Zhang Zhao, et al. Blockchain Technology: Architecture and progress [J]. Chinese Journal of Computer Science, 2018, 41(5): 969-988.

[3] Vujičić D, Jagodić D, Ranđić S. Blockchain technology, bitcoin, and ethereum: A brief overview[C]//17th International Symposium INFOTEH-JAHORINA (INFOTEH), 2018: 1-6.

[4] Buterin V, et al. A next-generation smart contract and decentralized application platform[J]. White Paper, 2014, 3(37): 2-1.

[5] Su L, Shen X, Du X, et al. Evil under the sun: Understanding and discovering attacks on ethereum decentralized applications[C]//USENIX Security Symposium, 2021: 1307-1324.

[6] Kaur G, Habibi Lashkari A, Sharafaldin I, et al. Introduction to smart contracts and defi[M]. Springer, 2023: 29-56.

[7] Zhou H, Milani Fard A, Makanju A. The state of ethereum smart contracts security: Vulnerabilities, countermeasures, and tool support[J]. Journal of Cybersecurity and Privacy, 2022, 2(2): 358-378.

[8] Huang Y, Bian Y, Li R, et al. Smart contract security: A software lifecycle perspective[J]. IEEE Access, 2019, 7: 150184-150202.

[9] Liao Z, Zheng Z, Chen X, et al. Smartdagger: a bytecode-based static analysis approach for detecting cross-contract vulnerability[C]//Proceedings of the 31st ACM SIGSOFT International Symposium on Software Testing and Analysis, 2022: 752-764.

[10] Wu K. An empirical study of blockchain-based decentralized applications[J]. arXiv preprint arXiv:1902.04969, 2019.

[11] Colombo C, Ellul J, Pace G J. Contracts over smart contracts: Recovering from violations dynamically[C]//Leveraging Applications of Formal Methods, Verification and Validation. Industrial Practice: 8th International Symposium, ISoLA 2018, Limassol, Cyprus, November 5-9, 2018, Proceedings, Part IV 8, 2018: 300-315.

[12] Li W, Bu J, Li X, et al. A survey of defi security: Challenges and opportunities[J]. arXiv preprint arXiv:2206.11821, 2022.

[13] Li W, Bu J, Li X, et al. Security analysis of defi: Vulnerabilities, attacks and advances[C]//2022 IEEE International Conference on Blockchain (Blockchain), 2022: 488-493.

[14] Zhang Y, Ma S, Li J, et al. Smartshield: Automatic smart contract protection made easy[C]//2020 IEEE 27th International Conference on Software Analysis, Evolution and Reengineering (SANER), 2020: 23-34.

[15] Kushwaha S S, Joshi S, Singh D, et al. Ethereum smart contract analysis tools: A systematic review[J]. IEEE Access,

2022, 10: 57037-57062.

[16] Luu L, Chu D H, Olickel H, et al. Making smart contracts smarter[C]//Proceedings of the 2016 ACM SIGSAC conference on computer and communications security, 2016: 254-269.

[17] Tsankov P, Dan A, Drachsler-Cohen D, et al. Securify: Practical security analysis of smart contracts[C]//Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, 2018: 67-82.

[18] Jiang B, Liu Y, Chan W K. Contractfuzzer: Fuzzing smart contracts for vulnerability detection[C]//2018 33rd IEEE/ACM International Conference on Automated Software Engineering (ASE), 2018: 259-269.

[19] Sürücü O, Yeprem U, Wilkinson C, et al. A survey on ethereum smart contract vulnerability detection using machine learning[J]. Disruptive Technologies in Information Sciences VI, 2022, 12117: 110-121.

[20] Tann W J W, Han X J, Gupta S S, et al. Towards safer smart contracts: A sequence learning approach to detecting security threats[J]. arXiv preprint arXiv:1811.06632, 2018.

[21] Zhuang Y, Liu Z, Qian P, et al. Smart contract vulnerability detection using graph neural network[C]//IJCAI, 2020: 3283-3290.

[22] Trozze A, Kleinberg B, Davies T. Detecting defi securities violations from token smart contract code with random forest classification[J]. arXiv preprint arXiv:2112.02731, 2021.

[23] Guo D, Dong J, Wang K. Graph structure and statistical properties of ethereum transaction relationships[J]. Information Sciences, 2019, 492: 58-71.

[24] Wu S, Wang D, He J, et al. Defiranger: Detecting price manipulation attacks on defi applications[J]. arXiv preprint arXiv:2104.15068, 2021.

[25] Zhou L, Qin K, Cully A, et al. On the just-in-time discovery of profit-generating transactions in defi protocols[C]//2021 IEEE Symposium on Security and Privacy (SP), 2021: 919-936.

[26] Wang D, Wu S, Lin Z, et al. Towards a first step to understand flash loan and its applications in defi ecosystem[C]//Proceedings of the Ninth International Workshop on Security in Blockchain and Cloud Computing, 2021: 23-28.

[27] Wang S H, Wu C C, Liang Y C, et al. Promutator: Detecting vulnerable price oracles in defi by mutated transactions[C]//2021 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), 2021: 380-385.

[28] Cao Y, Zou C, Cheng X. Flashot: a snapshot of flash loan attack on defi ecosystem[J]. arXiv preprint arXiv:2102.00626, 2021.

[29] Wang B, Liu H, Liu C, et al. Blockeye: Hunting for defi attacks on blockchain[C]//2021 IEEE/ACM 43rd International Conference on Software Engineering: Companion Proceedings (ICSE-Companion), 2021: 17-20.

[30] El-Dosuky M A, Eladl G H. Doorchain: deep ontology-based operation research to detect malicious smart contracts[C]//World Conference on Information Systems and Technologies, 2019: 538-545.

[31] Kehrli J. Blockchain 2.0-from bitcoin transactions to smart contract applications[EB/OL]. (2016-01-05)[2018-01-05]. https://www.niceideas.ch/roller2/badtrash/entry/blockchain-2-0-frombitcoin.

[32] von Haller Gronbaek M. Blockchain 2.0, smart contracts and challenges[J]. Comput. Law, SCL Mag, 2016: 1-5.

[33] Mohammed A H, Abdulateef A A, Abdulateef I A. Hyperledger, ethereum and blockchain technology: A short overview[C]//2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), 2021: 1-6.

[34] Chen T, Li Z, Zhu Y, et al. Understanding ethereum via graph analysis[J]. ACM Transactions on Internet Technology

(TOIT), 2020, 20(2): 1-32.

[35] Pierro G A, Rocha H. The influence factors on ethereum transaction fees[C]//2019 IEEE/ACM 2nd International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB), 2019: 24-31.

[36] Laurent A, Brotcorne L, Fortz B. Transactions fees optimization in the ethereum blockchain[J]. Blockchain: Research and Applications, 2022, 100074.

[37] Hirai Y. Defining the ethereum virtual machine for interactive theorem provers[C]//Financial Cryptography and Data Security: FC 2017 International Workshops, WAHC, BITCOIN, VOTING, WTSC, and TA, Sliema, Malta, April 7, 2017, Revised Selected Papers 21, 2017: 520-535.

[38] Fu Y, Ren M, Ma F, et al. Evmfuzz: Differential fuzz testing of ethereum virtual machine[J]. arXiv preprint arXiv:1903.08483, 2019.

[39] Wood G, et al. Ethereum: A secure decentralised generalised transaction ledger[J]. Ethereum project yellow paper, 2014, 151(2014): 1-32.

[40] Wohrer M, Zdun U. Smart contracts: security patterns in the ethereum ecosystem and solidity[C]//2018 International Workshop on Blockchain Oriented Software Engineering (IWBOSE), 2018: 2-8.

[41] Xie J, Tang H, Huang T, et al. A survey of blockchain technology applied to smart cities: Research issues and challenges[J]. IEEE Communications Surveys Tutorials, 2019, 21(3): 2794-2830.

[42] Hurlburt G. Might the blockchain outlive bitcoin?[J]. IT Professional, 2016, 18(2): 12-16.

[43] Kosba A, Miller A, Shi E, et al. Hawk: The blockchain model of cryptography and privacy-preserving smart contracts[C]//2016 IEEE symposium on security and privacy (SP), 2016: 839-858.

[44] Zhu X. Research on blockchain consensus mechanism and implementation[C]//IOP Conference Series: Materials Science and Engineering, 2019: 042058.

[45] Lasla N, Al-Sahan L, Abdallah M, et al. Green-pow: An energy-efficient blockchain proof-of-work consensus algorithm[J]. Computer Networks, 2022, 214: 109118.

[46] Kapengut E, Mizrach B. An event study of the ethereum transition to proof-of-stake[J]. arXiv preprint arXiv:2210.13655, 2022.

[47] Malla T B, Bhattarai A, Parajuli A, et al. Status, challenges and future directions of blockchain technology in power system: A state of art review[J]. Energies, 2022, 15(22): 8571.

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

[49] Leal F, Chis A E, González-Vélez H. Performance evaluation of private ethereum networks[J]. SN Computer Science, 2020, 1(5): 1-17.

[50] Xu X, Weber I, Staples M, et al. A taxonomy of blockchain-based systems for architecture design[C]//2017 IEEE international conference on software architecture (ICSA), 2017: 243-252.

[51] Wohrer M, Zdun U. Smart contracts: security patterns in the ethereum ecosystem and solidity[C]//2018 International Workshop on Blockchain Oriented Software Engineering (IWBOSE), 2018: 2-8.

[52] Fröwis M, Fuchs A, Böhme R. Detecting token systems on ethereum[C]//International conference on financial cryptography and data security, 2019: 93-112.

[53] Chen T, Zhang Y, Li Z, et al. Tokenscope: Automatically detecting inconsistent behaviors of cryptocurrency tokens in ethereum[C]//Proceedings of the 2019 ACM SIGSAC conference on computer and communications security, 2019: 1503-1520.

[54]Dapp industry report 2022[EB/OL]. (2022)[2022-12-31]. https://dappradar.com/blog/dapp-industry-report-2022-dapp-industry-proves-resilient-in-crypto-winter.

[55] Lee M R, Yen D C, Hurlburt G F. Financial technologies and applications[J]. IT Professional, 2018, 20(2): 27-33.

[56] Wang Z, Qin K, Minh D V, et al. Speculative multipliers on defi: Quantifying on-chain leverage risks[J]. Financial Cryptography and Data Security, 2022.

[57] Li J. Defi as an information aggregator[C]//Financial Cryptography and Data Security. FC 2021 International Workshops, Berlin, Heidelberg, 2021: 171-176.

[58] Qi M, Xu Z, Wang Z, et al. Deda: A defi-enabled data sharing and trading system[C]//Proceedings of the Fourth ACM International Symposium on Blockchain and Secure Critical Infrastructure, 2022: 47-57.

[59] Behnke R. Explained: The starstream and agora hack (april 2022)[EB/OL]. (2022-04)[2022-12-31]. https://www.halborn.com/blog/post/explained-the-starstream-and-agora-hack-april-2022.

[60] Chen J, Xia X, Lo D, et al. Defining smart contract defects on ethereum[J]. IEEE Transactions on Software Engineering, 2022, 48(1): 327-345.

[61] Werner S M, Perez D, Gudgeon L, et al. Sok: Decentralized finance (defi)[J]. CoRR, 2021, abs/2101.08778.

ABOUT THE AUTHOR

Teng Yuan was born in Jiangsu Province, China. He is pursuing a master's degree at Nanjing University of Posts and Telecommunications. His research includes web, blockchain, and DeFi.

E-mail: yuant6666@126.com