Phishing Scam Detection on Ethereum: Towards Financial Security for Blockchain Ecosystem

Weili Chen[1,2] , **Xiongfeng Guo** [1,3] , **Zhiguang Chen** [1,2] , **Zibin Zheng** [1,3] and **Yutong Lu** [1,2]
[1]School of Data and Computer Science, Sun Yat-sen University
[2]National Supercomputer Center in Guangzhou, Sun Yat-sen University
[3]National Engineering Research Center of Digital Life, Sun Yat-sen University
chenwli28@mail.sysu.edu.cn, guoxf6@mail2.sysu.edu.cn, zhiguang.chen@nscc-gz.cn,
zhzibin@mail.sysu.edu.cn, yutong.lu@nscc-gz.cn

Abstract

In recent years, blockchain technology has created a new cryptocurrency world and has attracted a lot of attention. It also is rampant with various scams. For example, phishing scams have grabbed a lot of money and have become an important threat to users' financial security in the blockchain ecosystem. To help deal with this issue, this paper proposes a systematic approach to detect phishing accounts based on blockchain transactions and take Ethereum as an example to verify its effectiveness. Specifically, we propose a graph-based cascade feature extraction method based on transaction records and a lightGBM-based Dual-sampling Ensemble algorithm to build the identification model. Extensive experiments show that the proposed algorithm can effectively identify phishing scams.

1 Introduction

The birth of Bitcoin has brought a whole new world of cryptocurrency. According to coinmarketcap.com, there are now over 5,000 cryptocurrencies (or tokens) with a market capitalization larger than $200 billion (see [Chen *et al.*, 2020] for a detailed analysis of the token market). The key technology behind these cryptocurrencies is blockchain technology. Generally speaking, a blockchain can be described as a distributed and trusted database maintained by a peer-to-peer network through a special consensus mechanism [Zheng *et al.*, 2018]. A blockchain usually implements a cryptocurrency (or a virtual currency) and it can be exchanged with other cryptocurrencies or fiat money through *exchanges*. The financial nature of cryptocurrency makes it the target of many scams.

Financial security is an important foundation for the healthy development of blockchain technology. The proliferation of scams in the ecosystem will hinder users' acceptance and use of blockchain technology, and further, hinder the progress of the technology. Thus, identification of these scams has become an urgent and critical problem in the blockchain ecosystem and has attracted great attention from researchers [Bartoletti *et al.*, 2020; Chen *et al.*, 2018]. The phishing scam is a new type of cybercrime that arises along with the rise of online business [Liu and Ye, 2001], which has now been found in the blockchain ecosystem. According to the report of *Chainalysis*, more than 50% of all cybercrime revenue was generated from phishing scams since 2017[1]. A widely known example is the phishing scam on Bee Token ICO [2], in which the phisher eventually gathered about $1 million from the investors in only 25 hours. These examples show that detecting and preventing phishing scams is an urgent problem in the blockchain ecosystem.

Traditional phishing scams typically involve setting up a fake official website and luring users into logging in to obtain private information, such as passwords. Thus, the main task of the traditional phishing scam detection method is to identify fake websites through various methods so that users can get an early warning before logging in. However, phishing scams in the blockchain era have many new characteristics. First of all, instead of private information, cryptocurrencies become the phishing targets. Phishers use a variety of methods to lure ordinary users to transfer money to a designated account (such as in the case of Bee Token ICO scam). Second, the ill-gotten cryptocurrencies have to be cashed through exchanges for fiat money (i.e., to convert the ill-gotten cryptocurrencies into fiat money) through transactions. Third, the transaction records of public blockchain are publicly accessible, which provides a new data source for phishing detection.

Based on these new characteristics and the fact that phishing scams are rampant in the blockchain ecosystem, we propose to build phishing scam detection methods based on blockchain transactions and AI. These methods can be incorporated into users' cryptocurrency wallets (i.e., tools for management of accounts and transactions in the blockchain ecosystem) as a function of alerting users to potential risks when interacting with unfamiliar accounts. Figure 1 shows the proposed framework and uses Ethereum as an example to demonstrate the effectiveness of our approach. Specifically, we first downloaded the Ethereum ledger using an Ethereum client Parity and crawled etherscan.io to get all the phishing accounts. Then, based on common sense and data analysis, we propose several filtering rules to alleviate the class imbalance problem. On this basis, we construct the transaction graph and propose a graph-based cascade feature extraction

[1] https://blog.chainalysis.com/the-rise-of-cybercrime-on-ethereum/

[2] https://theripplecryptocurrency.com/bee-token-scam/

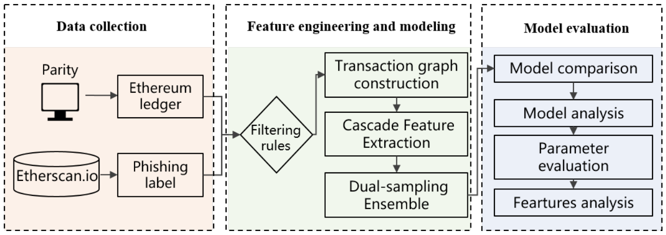

Figure 1: The framework.

method. Next, a Dual-sampling Ensemble framework is proposed to identify suspect accounts. Finally, we verify the validity of the model by comparing it with other methods, evaluate the performance of the model under different parameters, and discuss the effectiveness of these features.

In summary, we make the following major contributions.

(1) We propose a systematic approach to detect phishing scams in the blockchain ecosystem, and take Ethereum as an example to verify the effectiveness. The approach has good performance, which indicates that our method can be embedded into users' cryptocurrency wallets to provide users with a financial risk warning function. To accelerate the research in this field and promote the healthy development of blockchain technology, all relevant data and code will be released after the paper is published.

(2) We propose a graph-based cascade feature extraction method, which can conveniently extract rich transaction structure information and form a feature set with a good classification effect. Besides, it is very scalable and hard to evade according to the "six-degree separation" theorem.

(3) We propose a new model integration algorithm, namely the Dual-sampling Ensemble algorithm, which can be used for classification problems with a high level of class imbalance. The evaluation results show the effectiveness of the algorithm.

2 Background and Related Work

Blockchain technology is a key support technology for cryptocurrencies such as Bitcoin[3]. A blockchain can be seen as a common ledger maintained between peers that do not need to trust each other [Zheng *et al.*, 2017]. The ledger records the number of users' cryptocurrency and the history of transfer transactions between them. The user is represented in the system as a public-private key pair. Public keys, often called *addresses*, are like *accounts* in a banking system that records the cryptocurrency they hold. (In this paper, we use the term address and account interchangeably.) In blockchain systems, transactions are messages sending from one account (the initiator's address) to another (the receiver's address) [Chen *et al.*, 2018]. Typically, the initiator transfers a certain amount of cryptocurrency to the recipient. Transactions that occur over a period are packaged into blocks by peers and linked to the previous block through cryptography. Each block has a corresponding height (denoted as *blockNumber* in this paper), ~~increasing by 1 from 0. The block height can be viewed as the time when the transac~~tion took place. In the bitcoin system, blocks are created roughly every ten minutes. Ethereum is known as the second-generation blockchain technology because it provides full support for smart contracts [Wood, 2014]. A smart contract on a blockchain can be viewed as a piece of code that automatically executes and cannot be terminated when a given condition is met. Ethereum is now the largest platform for blockchain smart contracts and one of the main targets of various cyber attacks in the blockchain ecosystem. The cryptocurrency maintained by Ethereum is called *ether*.

In recent years, with the development of blockchain technology, financial security in the blockchain ecosystem has received extensive attention, and the identification of various fraudulent behaviors has become a research hotspot. In the Bitcoin ecosystem, [Vasek and Moore, 2015] presents the first empirical analysis of Bitcoin-based scams. The authors identify 192 scams and point out that at least 13,000 distinct victims lost more than $11 million. [Vasek and Moore, 2018] analyzes the supply and demand for Bitcoin-based Ponzi schemes, while [Bartoletti *et al.*, 2018] establish an address identification model for Ponzi scheme in the Bitcoin ecosystem. Besides, [Chen *et al.*, 2019a] show that there are market manipulation in the Bitcoin exchange Mt. Gox. In the Ethereum ecosystem, on the one hand, people are concerned with the identification of various scams, for example, smart Ponzi schemes [Bartoletti *et al.*, 2020; Chen *et al.*, 2018]. On the other hand, since most smart contracts control certain digital assets, ensuring that there are no vulnerabilities in the smart contracts is an important part of Ethereum's financial security [Kalra *et al.*, 2018].

Phishing detection has been extensively studied in the past decades and many methods have been proposed [Khonji *et al.*, 2013; Abdelhamid *et al.*, 2014; Zouina and Outtaj, 2017]. However, there is seldom research on phishing fraud identification considering the characteristics of blockchain. [Andryukhin, 2019] classify the main types and schemes of phishing attacks on the blockchain project and suggest methods of protection against phishing attacks from the blockchain project side's perspective. Unlike them, we are targeting the entire blockchain ecosystem and providing users with an early warning against phishing scams.

3 Proposed Method

Identifying phishing accounts in the blockchain system faces two challenges: 1) we only have transaction records and know little about account functions and holder information and 2) the number of phishing addresses is very few and other addresses are huge, identifying such a small group of accounts in the huge account set is like looking for a needle in the haystack. (The details of the data are described in Section 4.) To meet the challenges, the proposed method includes two parts, the cascade feature extraction method, and the lightGBM-based Dual-sampling Ensemble algorithm.

3.1 Cascade Feature Extraction Method

Since transaction records are the only information we can use, and they give the accounts a natural graphical structure,

[3] https://bitcoin.org/bitcoin.pdf

to extract effective features, we first construct a transaction graph (TG) based on these transaction records. Specifically, $TG = (V, E)$, where V is a set of nodes (all the addresses in the dataset) and $E = \{(v_i, v_j)|v_i, v_j \in V\}$ is a set of ordered edges. Each edge indicates that an address V_i transfers a certain amount of ether to another address V_j. Each edge has two attributes: *blockNumber* and *amount*, representing the time when this edge emerges and the amount of the transaction. Please note that there may be multiple edges between two nodes in TG, depending on the number of transactions between the two related accounts. (we use account, address, and node interchangeably in the following.) Next, we introduce the proposed feature extraction method.

Graph-based features have proven to be very effective in many identification problems [Chatzakou *et al.*, 2017; Ramalingam and Chinnaiah, 2018]. Thus, we propose a TG-based *cascade feature extraction* method for phishing account identification. The idea is as follows. Treat the transaction between accounts as a *friend* relationship, to judge the category of an account, we can use not only the information of the account, but also the information of its friends, even the information of its friends' friends, and so on. To explain more clearly, we first define several keywords related to a node.

- *Node data*: Node data is the transaction history of that node. Each transaction contains information about the time, direction, and amount of the transaction. The transaction time is denoted as *blockNumber*, which is an increasing integer. A transaction has two directions: *out* and *in*. The out-transactions of an account transfer ether from the account to other accounts and the in-transactions of an account receive ether from other accounts.
- *Node features*: Node features are all kinds of information extracted from node data. In this paper, we extract information through various statistical methods.
- *N-order friend*: A node's 1-order friend is a node directly connected to the node (i.e., there are transactions between them). A node's n-order friend is a node connected to the node with at least n-1 nodes.
- *N-order features*: The 0-order features of a node is the node features of that node. The n-order features are extracted in cascade from the n-order friends.

To explain how to achieve cascade feature extraction, we show the procedure of 2-order features extraction in Figure 2. Suppose we need to compute the 2-order features of node *A*, which have 1-order friends *B, C* and 2-order friends *D, E, F, G, H*. In the figure, each undirected edge represents one or more transactions (regardless of the directions) between two nodes, and the counterparty of the 2-order friends is not shown. The procedure is divided into three stages. In the first stage, we compute a statistic (i.e., the grey rectangle) for each 2-order friends by using its node data (i.e., the transaction history). The second stage needs to calculate a statistic for each 1-order friend by using the statistics computed in the first stage (not the node data of the 1-order friend). Similarly, in the last stage, we still calculate a statistic whose input comes from the second stage. This approach is very scalable. In fact, by increasing the order and using different statistic methods at different stages, we can extract rich information about how a node interacts with the entire network. It should be noted that the approach we describe here does not take into account the direction of the transaction. But, for phishing accounts, in-transactions and out-transactions are significantly different in meaning. Therefore, in this paper, we extract features for two different directions respectively.

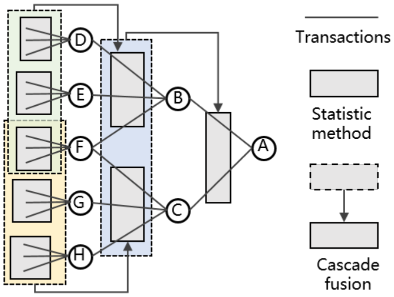

Figure 2: Example of 2-order feature extraction procedure.

Node Features

The node features are statistics of its node data. There are two *types* of data: transaction amount and transaction times (i.e., *block*Number). In order to distinguish the nature of the transaction, statistics are made in different *directions* (i.e., *out*-transactions or *in*-transactions). For convenience, we name these features as *direction_type_method*. For example, a feature *in_block_std* of a node indicates the standard deviation (i.e., the method *sd*) for the transaction time (i.e., the type of data *block*) of all in-transactions (i.e., the transaction direction *in*). For the transaction time, we compute only the transaction time span (denoted as *ptp*) and its standard deviation (denoted as *sd*). For the transaction amount, we calculated the sum, the maximum, the minimum, the mean and the standard deviation (i.e., *sd*). In addition, there are statistics unrelated to transaction amount: *count*, *unique*, and *unique_ratio*. They represent the number of transactions (i.e., *count*), the number of counterparties (i.e., *unique*), and the ratio of the two (i.e., $unique/count$). By doing so, we obtained 19 features (i.e., $2 \times 1 \times (2 + 5 + 2) + 1$).

N-order Features

For simplicity, in this study, we extract only 1-order network features. As mentioned, the direction of the transaction is important in identifying phishing scams. Thus, considering the transaction direction, the 1-order friends of a node can be divided into *from* friends and *to* friends. In simple terms, when there is a transfer transaction from node *A* to node *B*, we call node *B* a *from* friend of node *A* and node *A* a *to* friend of node *B*. Specifically, the 1-order network features are named as *friend_direction_statistic2_statistic1*. For example, the *from_in_mean_max* feature is calculated as follows: we first compute the maximum (i.e., *max*) of the *in*-transaction amounts for each *from* friend. Then, we compute the *mean* of all statistics in the previous stage. Similarly, to compute *to_out_std_sum*, we first compute the *sum* of all the *out*-transaction amounts for each *to* friend. Then, we compute the standard deviation (i.e., *sd*) of all statistics in the previous stage. By doing so, we can obtain 200 features (i.e., $2 \times 2 \times 2 \times 5 \times 5$). Please note that we did not take time into account in the 1-order network feature extraction.

3.2 Dual-sampling Ensemble Method

Identifying phishing scams is essentially establishing a classification model of addresses. But the phishing account identification faces a class imbalance problem. To build a useful suspect identification model, we propose a Dual-sampling Ensemble method, an identification framework integrated with many base models trained by sampling examples and features.

Base Model

The base models play a central role in the identification framework. Many mature classification algorithms can be used as base models, such as logistic regression (LR), support vector machine (SVM), and decision tree (DT). Among these models, the gradient boosting decision tree (GBDT) obtained good results in many problems. There are several different variants of GBDT, including XGBoost [Chen and Guestrin, 2016] and lightGBM [Ke *et al.*, 2017], which are widely used and generally accepted. In the phishing detection problem, we found that lightGBM is more efficient, thus we select it as our base model.

Given the supervised training set $X = \{(x_i, y_i), i = 1, 2, \cdots, n\}$, lightGBM integrates a number of K regression trees $f(x) = \frac{1}{K}\sum_{i=1}^{T} h_i(x)$ to approximate a certain function $f^*(x)$ that minimizes the expected value of a specific loss function $L(y, f(x))$. In each iteration of GBDT, assume that the strong learner obtained by the previous iteration is $h_{t-1}(x)$, the loss function is $L(f(x), h_{t-1}(x))$, then the aim for the current iteration is to find a week learner using CART regression tree model which denoted as $h_t(x)$, to minimize the formula $L(f(x), h_{t-1}(x) + h_t(x))$. Suppose in iteration t, the negative gradient for sample i can be represented as $r_{ti} = \frac{\partial L(y_i, h_{t-1}(x_i))}{\partial h_{t-1}(x_i)}$. By using the Log-likelihood loss as loss function$L(y, h(x)) = log(1 + exp(-yh(x)))$, where $y \in [-1, 1]$, we can simplify the negative gradient of sample as below:

$$r_{ti} = -\frac{\partial L(y_i, h_{t-1}(x_i))}{\partial h_{t-1}(x_i)} = \frac{y_i}{1 + exp(y_i h(x_i))},$$

where $i = 1, 2, \cdots, m$.

By using the formula, LightGBM chooses to remove these small gradient samples from the training set to make the model pay more attention to those samples which cause great Loss. This technique is called Gradient-based One-Side Sampling (GOSS) [Ke *et al.*, 2017]. When constructing the CART regression tree, LightGBM binds the mutual exclusion features so that the number of features (the leaves) can be greatly reduced.

Dual-sampling Ensemble

Inspired by *EasyEnsemble* [Liu *et al.*, 2008], we propose a Dual-sampling Ensemble algorithm to solve the class imbalance problem in the phishing scam identification. The pseudocode is shown in Algorithm 1.

Algorithm 1 The Dual-sampling Ensemble algorithm

Input: The minority class example set $\mathcal{P}$, the majority example set $\mathcal{N}$,$|\mathcal{P}| \ll |\mathcal{N}|$, the number of base models k, the feature sample ratio r, and the number of features d, The best parameters for the base model
Output: The integration result.

```
Let $i \leftarrow 0$;
while i $<$ k do
    $i \leftarrow i + 1$;
    Randomly sample a subset $N_i$ from $N$, $|N_i| = \lfloor \frac{N}{K} \rfloor$;
    Learn a base model $h_i$ using $\mathcal{P} \cup \mathcal{N}_i$ with only $d \times r$ randomly sampled features. The parameters are sampled around the best parameters;
end while
return $H(x) = \frac{1}{K}\sum_{i=1}^{T} h_i(x)$
```

The idea behind the Dual-sampling Ensemble is simple. Similar to *EasyEnsemble* [Liu *et al.*, 2008], we reduce the class imbalance by sampling the majority example set (i.e., negative examples). The difference is that we also sample the features of the examples in the training set since we can obtain a large number of features by using the cascade feature extraction method. This dual sampling method allows the base models to have better heterogeneity.

4 Data Collection and Preparation

4.1 Data Collection

We launch an Ethereum client, Parity[4], on our server to download the ledger of Ethereum. By using Parity, we obtained all the Ethereum blocks before January 3, 2019 (to be exact, from block height 0 to block height 7,000,000). By analyzing the transactions obtained, we get 43,783,194 accounts, among which 1,564,580 accounts controlled by smart contracts.

One of the most important tasks in establishing a phishing scam identification model is to find enough phishing account examples. Fortunately, etherscan.io provides several tags for Ethereum addresses, and by crawling the website, we obtain all the addresses labeled with Phishing[5]. These addresses are used in some verified phishing scams. In this way, we obtain 1,683 phishing addresses. We call these phishing addresses as *positive* examples and the rest as *negative* examples.

4.2 Data Cleaning

After getting all the data, we found that the class was very imbalanced. The class imbalance ratio, i.e., the ratio of the size of the majority class (negative examples) to minority class (positive examples), exceeds 26,000. Given that some addresses are not phishing addresses, we recommend that some obvious negative examples (i.e., non-phishing addresses) be eliminated before model training in order to build a more effective model. To this end, we 1) filter transaction records involving a smart contract address, 2) eliminate addresses

[4] www.parity.io/ethereum/

[5] etherscan.io/accounts/label/phish-hack

with less than 10 or more than 1,000 transaction records, and 3) ignore all transactions that appear before block height 2 million.

The above cleaning methods are based on the following considerations. First of all, smart contracts often have complex logic and are not convenient for phishing scams. Furthermore, smart contracts account for very little in the phishing addresses (i.e., 2.6%), and they usually relate to tokens. Thus, In this preliminary study, for the sake of simplicity, we leave out smart contracts. Second, we want to learn the behavioral characteristics of phishing accounts through transaction records, and too few records are not good for learning. Besides, too many records indicate that the account may be a wallet or other type of accounts. In fact, there are many addresses (i.e., >70%) with more than 1,000 transaction records, and only one address is labeled with phishing. Finally, by analyzing the initial activity time of phishing addresses, we find that all phishing addresses are active after 2016-08-02. This may be because, in the early days of Ethereum, phishing scams were relatively few, and even fewer were recorded. Therefore, we proposed to build the model based on records after block height of 2 million (i.e., 2016-08-02). These filtering rules allow the model to focus on learning the characteristics of phishing scams.

5 Experiment Result and Analysis

5.1 Experiment Settings

We downloaded all of Ethereum's transaction data from its inception to January 3, 2019 (i.e., from block height 0 to block height 7,000,000). By using the filter rules in Section 4.2, we ended up with 7,795,044 transaction records. There are 534,820 addresses, 323 of which are phishing addresses. The following experiments are based on this data set. In order to reflect the effectiveness of the model more accurately and avoid the contingency caused by the partitioning of train and test sets, the paper adopts the evaluation method of k-fold cross-validation. Specifically, we set the parameter k=5. To accurately evaluate the model, we select four metrics: precision, recall, F1, and AUC, which is commonly used in classification problems.

5.2 Method Comparison

In order to verify that our proposed model is more suitable for this problem, we compared the single-model lightGBM, Support Vector Machine (SVM), decision tree (DT), and their Dual-sampling Ensemble (DE+) models. SVM and DT are considered efficient in many classification problems of class imbalance [Chen *et al.*, 2019b]. Thus, we chose it as the baseline of our model. To compare the performance of these methods, we set the feature sampling rate to 70%, and the number of base models to 1600 (i.e., balance ensemble). Table 1 shows the results. As can be seen, in these single-models, SVM performs poorly, lightGBM and DT have certain performance, but they are obviously of no practical value. On the contrary, after adopting the ensemble strategy, the performance of each model is significantly improved, especially lightGBM and DT (i.e., DElightGBM and DEDT). This result shows that the ensemble method is a good choice when facing the class imbalance. It is worth noting that the proposed model (i.e., DElightGBM) performs well on all metrics (i.e., all larger than 0.8). It means that the proposed model can be deployed in a real wallet for real-time warnings.

Method	Precision	Recall	F1	AUC
SVM	0.0000	0.0002	0.0000	0.4817
DT	0.0552	0.0810	0.0657	0.5630
lightGBM	0.0535	0.0745	0.0623	0.5364
DESVM	0.2222	0.0076	0.0146	0.5046
DEDT	0.7295	0.7167	0.7230	0.7183
DElightGBM	**0.8196**	**0.8050**	**0.8122**	**0.8097**

Table 1: The performance comparison

#models	Precision	Recall	F1	AUC
1	0.0789	0.0991	0.0879	0.549
100	0.7583	0.3993	0.5232	0.6947
800	**0.9288**	0.7368	**0.8217**	**0.8274**
1000	0.826	0.7585	0.7908	0.8206
1600	0.8196	**0.805**	0.805	0.8097

Table 2: The effect of example sampling (with lightGBM)

5.3 Example Sampling Effect Analysis

Evaluating the impact of example sampling on the model is essentially selecting the number of base models. Table 2 shows the four evaluation metrics of the framework DElightGBM with different numbers of base models. (We set the feature sampling rate to 70% and the parameters of each model are randomly selected around the optimal parameters.) It can be seen that with the increase in the number of base models, all the metrics obtained different degrees of promotion. When the number of base models reaches 800 (i.e., half balance ensemble), three metrics (i.e., *precision*, *F1* and *AUC*) reach the maximum. However, the *recall* keeps going up, and it reaches its maximum when the number of base models is 1600 (i.e., balance ensemble). This result indicates that the level of class imbalance is a very important factor affecting the performance of base models. From the experimental results, half balance ensemble seems to be a good choice. To make the model more practical, however, we would prefer to find all potential phishing scams (i.e., higher *recall*) at the expense of precision. Therefore, we propose the use of the balance ensemble for phishing scam detection.

5.4 Feature Sampling Evaluation

Next, we analyze the effect of feature sampling by setting different sampling ratios. To eliminate the effect of the number of base models, it is uniformly set at 1600. Table 3 shows the evaluation results. In general, the feature sampling method has a certain influence on the final results, however, as compared with example sampling, its influence is far less significant. From the perspective of the most preferred metric, *recall*, 0.8 is the best feature sampling ratio. Compared to using all the features (i.e., ratio=1), *recall* improved 4.24%.

Ratio	Precision	Recall	F1	AUC
0.6	0.8228	0.7832	0.8025	0.8018
0.7	0.8149	0.8205	0.8177	0.8127
0.8	**0.8258**	**0.8390**	**0.8324**	**0.8282**
0.9	0.8055	0.7955	0.8005	0.7957
1.0	0.8282	0.8049	0.8164	0.8096

Table 3: The effect of feature sampling

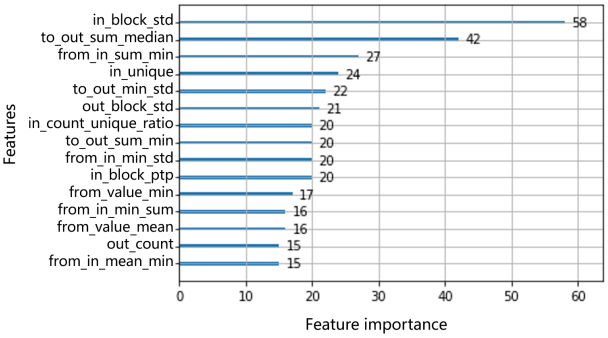

Figure 3: The top 15 important features.

These results reveal a noteworthy phenomenon. It is not necessarily correct that the more features the model has, the better the performance. On the contrary, in the case that we can obtain a large number of features, a certain degree of feature sampling is conducive to obtaining a better model. This may because feature sampling can make different base models view the object from different angles, so as to obtain better identification.

5.5 Feature Analysis

Since we adopted the method of cascading feature extraction, a large number of features were obtained. Figure 3 shows the top 15 important features in the model. Next, we analyze why some of these features are important.

- *in_block_std* is the standard deviation of *blockNumber* of all *in* transaction for a node. This feature reflects the intensity of in-transactions at a certain address. If there is a large number of in-transactions in a short period, the *blockNumber* of these transactions will be very close to each other, and thus the constructed *to_block_std* will be very small. This feature is much more important than the others, and its meaning is easy to understand. For a phishing address, a natural phenomenon is that the number of in-transactions increased suddenly within a period after the phishing began. However, with the phishing scam revealed, in-transactions become rare, or even nonexistent. This leads to in-transactions are concentrated in a small period for a phishing address, and the feature can grasp this characteristic very well.
- *to_out_sum_median* is a typical 1-order network feature. It reflects the overall situation (i.e., *sum*) of all the *to* friends' out-transactions. This feature is not as intuitive as the previous one and requires some explanation to understand its value. First of all, we can think of the median amount of *out* transaction of an address as an indicator of its financial strength. This is not difficult to understand, because the large median means that at least half of the address's *out* transaction amounts are large, indicating that its financial strength is stronger. Second, for phishing addresses, *to* friends are the victims of the phishing scam. Thus, for phishing scams, this feature can be seen as an indication of the overall financial strength of all its victims.
- *from_in_sum_min* is also an 1-order network feature. Different from the previous feature, this feature reflects the *in* transaction of the node's *from* friend. It is relatively easy to understand why the feature is important. For phishing scams, money laundering is an important part before cashing out. Therefore, the *from* friend of the phishing address, which is usually the intermediate address used for money laundering, must exhibit behavior characteristics different from normal addresses. And, this type of features captures the difference effectively.

The above analysis of the top three features shows that our feature engineering achieves good results, fully mining the characteristics of the node itself and different neighbors of the node.

6 Conclusion and Future Work

In blockchain ecosystems, various scams are rampant, which seriously threaten the financial security of users involved. To help dealing with this issue, in this study, we propose a systematic approach to detect phishing scams in the Ethereum ecosystem. First of all, by using the Parity client and crawl etherscan.io, we collect all transactions of the Etehreun blockchain and the labeled phishing addresses. Then, by using this data, we construct a transaction graph and propose a graph-based cascade feature extraction method, which helps us extract many useful features. Next, based on the extracted features and lightGBM, we propose a Dual-sampling Ensemble model to detect phishing suspects. Finally, we evaluate the model from many angles, and the results indicate the effectiveness of our model. In the future, we are going to further this study to other cybercrimes and set up a blockchain scam detection website to provide the phishing scam identification service in the form of API. Besides, to accelerate the research in this field, all relevant data and code will be released after the paper is published.

Acknowledgments

The work described in this paper was supported by the National Key R&D Program of China (2018YFB0204303), the National Natural Science Foundation of China (61722214), the Natural Science Foundation of Guangdong (2018B030312002, 2019B020214006), China Postdoctoral Science Foundation (Grant no. 2019TQ0372, 2019M660223), the Program for Guangdong Introducing Innovative and Entrepreneurial Teams under Grant NO. 2016ZT06D211 and Pearl River S&T Nova Program of Guangzhou under Grant NO. 201906010008. Zhiguang Chen and Zibin Zheng are the corresponding authors.

References

[Abdelhamid *et al.*, 2014] Neda Abdelhamid, Aladdin Ayesh, and Fadi Thabtah. Phishing detection based associative classification data mining. *Expert Systems with Applications*, 41(13):5948–5959, 2014.

[Andryukhin, 2019] AA Andryukhin. Phishing attacks and preventions in blockchain based projects. In *Proceedings of the International Conference on Engineering Technologies and Computer Science (EnT)*, pages 15–19. IEEE, 2019.

[Bartoletti *et al.*, 2018] Massimo Bartoletti, Barbara Pes, and Sergio Serusi. Data mining for detecting bitcoin ponzi schemes. In *Proceedings of the Crypto Valley Conference on Blockchain Technology*, pages 75–84. IEEE, 2018.

[Bartoletti *et al.*, 2020] Massimo Bartoletti, Salvatore Carta, Tiziana Cimoli, and Roberto Saia. Dissecting ponzi schemes on ethereum: Identification, analysis, and impact. *Future Generation Computer Systems*, 102:259–277, 2020.

[Chatzakou *et al.*, 2017] Despoina Chatzakou, Nicolas Kourtellis, Jeremy Blackburn, Emiliano De Cristofaro, Gianluca Stringhini, and Athena Vakali. Mean birds: Detecting aggression and bullying on twitter. In *Proceedings of the ACM on web science conference*, pages 13–22. ACM, 2017.

[Chen and Guestrin, 2016] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794. ACM, 2016.

[Chen *et al.*, 2018] Weili Chen, Zibin Zheng, Jiahui Cui, Edith Ngai, Peilin Zheng, and Yuren Zhou. Detecting ponzi schemes on ethereum: Towards healthier blockchain technology. In *Proceedings of the World Wide Web Conference (WWW2018)*, pages 1409–1418. International World Wide Web Conferences Steering Committee, 2018.

[Chen *et al.*, 2019a] Weili Chen, Jun Wu, Zibin Zheng, Chuan Chen, and Yuren Zhou. Market manipulation of bitcoin: Evidence from mining the mt. gox transaction network. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pages 964–972. IEEE, 2019.

[Chen *et al.*, 2019b] Weili Chen, Zibin Zheng, Edith C-H Ngai, Peilin Zheng, and Yuren Zhou. Exploiting blockchain data to detect smart ponzi schemes on ethereum. *IEEE Access*, 7:37575–37586, 2019.

[Chen *et al.*, 2020] Weili Chen, Tuo Zhang, Zhiguang Chen, Zibin Zheng, and Yutong Lu. Traveling the token world: A graph analysis of ethereum erc20 token ecosystem. In *Proceedings of the World Wide Web Conference (WWW2020)*, pages 1409–1418. International World Wide Web Conferences Steering Committee, 2020.

[Kalra *et al.*, 2018] Sukrit Kalra, Seep Goel, Mohan Dhawan, and Subodh Sharma. Zeus: Analyzing safety of smart contracts. In *Proceedings of the Network and Distributed System Security Symposium*, 2018.

[Ke *et al.*, 2017] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *Proceedings of the International Conference on Advances in Neural Information Processing Systems*, pages 3146–3154, 2017.

[Khonji *et al.*, 2013] Mahmoud Khonji, Youssef Iraqi, and Andrew Jones. Phishing detection: a literature survey. *IEEE Communications Surveys & Tutorials*, 15(4):2091–2121, 2013.

[Liu and Ye, 2001] Jiming Liu and Yiming Ye. Introduction to e-commerce agents: Marketplace solutions, security issues, and supply and demand. In *E-Commerce Agents*, pages 1–6. Springer, 2001.

[Liu *et al.*, 2008] Xu-Ying Liu, Jianxin Wu, and Zhi-Hua Zhou. Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(2):539–550, 2008.

[Ramalingam and Chinnaiah, 2018] Devakunchari Ramalingam and Valliyammai Chinnaiah. Fake profile detection techniques in large-scale online social networks: A comprehensive review. *Computers & Electrical Engineering*, 65:165–177, 2018.

[Vasek and Moore, 2015] Marie Vasek and Tyler Moore. There's no free lunch, even using bitcoin: Tracking the popularity and profits of virtual currency scams. In *Proceedings of the International Conference on Financial Cryptography and Data Security*, pages 44–61. Springer, 2015.

[Vasek and Moore, 2018] Marie Vasek and Tyler Moore. Analyzing the bitcoin ponzi scheme ecosystem. In *Proceedings of the International Conference on Financial Cryptography and Data Security*, pages 101–112. Springer, 2018.

[Wood, 2014] Gavin Wood. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum Yellow Paper*, 2014.

[Zheng *et al.*, 2017] Zibin Zheng, Shaoan Xie, Hongning Dai, Xiangping Chen, and Huaimin Wang. An overview of blockchain technology: Architecture, consensus, and future trends. In *Proceedings of the IEEE International Congress on Big Data*, pages 557–564. IEEE, 2017.

[Zheng *et al.*, 2018] Zibin Zheng, Shaoan Xie, Hongning Dai, Xiangping Chen, and Huaimin Wang. Blockchain challenges and opportunities: A survey. *International Journal of Web and Grid Services*, 14:352–375, 2018.

[Zouina and Outtaj, 2017] Mouad Zouina and Benaceur Outtaj. A novel lightweight url phishing detection system using svm and similarity index. *Human-centric Computing and Information Sciences*, 7(1):17, 2017.