Hindawi
Security and Communication Networks
Volume 2021, Article ID 6643763, 10 pages
https://doi.org/10.1155/2021/6643763

Research Article

Bitcoin Theft Detection Based on Supervised Machine Learning Algorithms

Binjie Chen,[1] **Fushan Wei**,[1] **and Chunxiang Gu**[1,2]

[1]*Henan Key Laboratory of Network Cryptography Technology, Zhengzhou 450001, China*
[2]*State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450001, China*

Correspondence should be addressed to Chunxiang Gu; gcx5209@126.com

Received 15 November 2020; Revised 12 January 2021; Accepted 19 January 2021; Published 26 February 2021

Academic Editor: Kim-Kwang Raymond Choo

Since its inception, Bitcoin has been subject to numerous thefts due to its enormous economic value. Hackers steal Bitcoin wallet keys to transfer Bitcoin from compromised users, causing huge economic losses to victims. To address the security threat of Bitcoin theft, supervised learning methods were used in this study to detect and provide warnings about Bitcoin theft events. To overcome the shortcomings of the existing work, more comprehensive features of Bitcoin transaction data were extracted, the unbalanced dataset was equalized, and five supervised methods—the k-nearest neighbor (KNN), support vector machine (SVM), random forest (RF), adaptive boosting (AdaBoost), and multi-layer perceptron (MLP) techniques—as well as three unsupervised methods—the local outlier factor (LOF), one-class support vector machine (OCSVM), and Mahalanobis distance-based approach (MDB)—were used for detection. The best performer among these algorithms was the RF algorithm, which achieved recall, precision, and $F1$ values of 95.9%. The experimental results showed that the designed features are more effective than the currently used ones. The results of the supervised methods were significantly better than those of the unsupervised methods, and the results of the supervised methods could be further improved after equalizing the training set.

1. Introduction

Blockchain is an integrated application of distributed data storage, peer-to-peer transmission, consensus mechanism, cryptographic algorithms, and other technologies and has the advantages of decentralization, data immutability, and programmability. Since Satoshi Nakamoto first proposed Bitcoin's concept [1] in 2008, the underlying technology blockchain has been widely used in the fields of digital currency, smart manufacturing, supply chain management, anticounterfeiting data service, and so on [1–7]. Simultaneously, blockchain security has also received widespread attention from the academic community.

Bitcoin is essentially a cryptographic digital currency and is by far the most successful blockchain application. As of October 2020, there were 7,378 cryptocurrencies worldwide with a total market capital of over $359.7 billion, of which Bitcoin accounted for approximately 58.3% of the market capital. Bitcoin uses blockchain technology to carry transaction records, creating a decentralized distributed ledger among Bitcoin users and enabling the creation, issuance, and trading of currency. Bitcoin enables global, low-cost instant money transfers and uses pseudonyms for privacy and anonymity. According to Blockchain.info, a real-time blockchain monitoring website, about $21 billion worth of transactions are now written into the Bitcoin blockchain every day on average, and more than 650,000 blocks have been created.

The anonymity and advantage of low-cost instant transfer and enormous economic value of Bitcoin have led to numerous related criminal incidents. These criminal incidents can be categorized into money laundering, use of extortion software, fraud, theft, and transactions on dark web markets. Among these, Bitcoin theft is the most destructive. In February 2014, Mt. Gox, once the largest Bitcoin trading platform in the world, announced that 850,000 Bitcoins, with a value of more than $450 million, might have been stolen and eventually went bankrupt. In August 2016, the Hong Kong exchange Bitfinex announced that it had suffered a security breach and that Bitcoins worth $72 million had been stolen from customer accounts. The price of Bitcoin fell 20% after the incident. Bitcoin theft events have huge impact on Bitcoin security and even socioeconomic security, making the ability to detect Bitcoin theft events and provide early warning in a timely manner of great theoretical value and practical significance. There have been many studies on the criminal incidents of Bitcoin and other public chain digital currencies in the existing literature. In 2016, Pham et al. [8, 9] extracted the features of the Bitcoin user graph and transaction graph, studied them from the perspective of the network using the power law and densification law, and detected 1 of 30 known Bitcoin abnormal events using three unsupervised methods: local outlier factor (LOF), one-class support vector machine (OCSVM), and Mahalanobis distance-based approach (MDB). In 2017, Toyoda et al. [10] analyzed the transaction pattern of Bitcoin addresses related to the high yield investment plan (HYIP). They extracted the features of known transaction patterns and marked the Bitcoin addresses. Through supervised learning, they classified more than 1,500 Bitcoin addresses, with a recall of 83% and a false positive rate (FPR) of 4.4%. In 2018, Vasek et al. [11] conducted a survival analysis on Ponzi schemes in Bitcoin to determine the factors affecting the persistence of fraud. They found 1,780 different Bitcoin Ponzi schemes by combing 1,424 posts on Bitcointalk and determined the positive correlation between the amount of interaction among the fraudsters and their victims and the duration of the scheme using survival analysis. In 2018, Chen et al. [12] used data mining and machine learning to detect Ponzi schemes in Ethereum. By examining Ethereum's smart contracts, extracting transaction features and code frequency characteristics from the accounts and opcodes of the smart contracts, and using eXtreme Gradient Boosting (XGBoost) to build detection models, 45 smart contracts of Ethereum that carried out Ponzi schemes were identified. They further estimated that there were more than 400 Ponzi schemes on Ethereum. In 2019, Torres et al. [13] tested the new fraudulent behavior Honeypot in Ethereum. Based on Honeypot's taxonomy, they built a method called HONEYBADGER that uses symbolic execution and heuristics to automatically detect Honeypot fraud. They conducted a large-scale analysis of more than 2 million smart contracts, identifying 690 Honeypot contracts and 240 victims. In 2019, Chen et al. [14] revealed the market manipulation phenomenon of digital currency through mining the trading network of digital currency exchange. They took the trading history of the Mt. Gox Bitcoin Exchange as a sample, divided it into three categories according to the characteristics of the accounts and constructed the trading history into three diagrams. On this basis, they identified the basic network with high correlation with price fluctuation through matrix singular value decomposition (SVD) and further found a large number of market manipulation patterns. In the same year, they [15] used an improved Apriori algorithm to detect users in the digital currency market who might be involved in a pump and dump scam. After analyzing the trading history of Mt. Gox Bitcoin Exchange, they found a large number of users who bought and sold at the same time, as well as abnormal trading behaviors and trading prices associated with them. Further analysis indicated that these users may be involved in the pump and dump scam. In 2019, Yang et al. [16] analyzed the characteristics of Bitcoin transaction data. They used the Gaussian mixture model (GMM) for clustering and detected a known theft event through the analysis of the clustering results. In 2020, Bartoletti et al. [17] systematically studied Ponzi schemes on Ethereum, collected a set of Ponzi schemes, and analyzed them. They examined contracts with specific source code and searched in Google to determine whether a Ponzi scheme had been committed, and they further extended existing collections by searching in blockchain based on bytecode similarity. On this basis, they studied the contract code pattern, trading volume, time behavior, and user characteristics of Ponzi schemes.

The above works have positive effects on reducing the occurrence of abnormal events of digital currency. However, the current studies on the detection of the destructive abnormal type-Bitcoin theft are far from mature and the existing studies involving this issue have the following deficiencies: (1) the features extracted from the user and transaction graphs were relatively simple and (2) unsupervised algorithms were generally used for detection. To improve upon the existing approaches, we extracted specific transaction features according to the characteristics of Bitcoin theft events. Further, we used five supervised methods—the k-nearest neighbor (KNN), support vector machine (SVM), random forest (RF), adaptive boosting (AdaBoost), and multilayer perceptron (MLP) approach—as well as three unsupervised methods—the LOF, OCSVM, and Mahalanobis distance-based algorithm—for detection. To improve the learning effects of these supervised methods, we equalized the imbalanced training data. Experiments showed that the KNN, RF, and AdaBoost algorithms achieved good results on our dataset, with *F*1 values exceeding 80%. In particular, the RF algorithm achieved recall, precision, and *F*1 values of 95.9%.

The remainder of this paper is organized as follows. Section 2 reviews the machine learning algorithms used in this study. Section 3 describes the dataset acquisition and feature extraction methods. Section 4 presents the experimental results and analysis. Finally, Section 5 summarizes the paper and provides an outlook for future work.

2. Preliminaries

In this study, we mainly used five supervised learning methods, and we will briefly review these methods in this section. In the comparison experiment, we used three unsupervised learning methods as well, which are not introduced here because they are not the focus of this article.

2.1. KNN.

The k-nearest neighbor (KNN) method [18] belongs to instance-based learning. The difference between instance-based learning and model-based learning is that the former does not require training or parameter tuning and can be used to make predictions directly.

We assume that the training set is D and k is the initial value of the number of the nearest neighbors. There is no convenient way to determine the best value of k directly, and its best value varies significantly between fields, so we generally set an initial value and adjust it according to the experimental results. For sample x_q to be classified, $x_1, \ldots, x_k$ denote the k-nearest samples in D. $V = \{v_1, \ldots, v_s\}$ denote s classes, and we obtain the category of x_q:

$$F(x_q) = \arg\max_{v \in V} \sum_{i=1}^{k} \delta(v, f(x_i)), \tag{1}$$

where when $a = b$, $\delta(a, b) = 1$; otherwise, $\delta(a, b) = 0$.

2.2. SVM.

2.2. SVM. The support vector machine (SVM) method [19] is a type of model-based learning and is one of the most powerful classifiers at present. The basic idea is to divide the data into two parts using a hyperplane, with the closest distance between the hyperplane and data points (we refer this distance as classification interval in the following text) reaching the maximum value.

We assume that the training set is $D = \{(x_i, y_i) | i = 1, 2, \ldots, m, x \in R^n, y \in \{\pm 1\}\}$ and that the classification hyperplane is $(w \cdot x) + b = 0$. To enable the hyperplane to correctly classify all samples, it is necessary to satisfy the following constraints: $y_i[(w \cdot x_i) + b] \geq 1, \quad i = 1, 2, \ldots, m$.

In addition, when the classification interval reaches its maximum, the closest distances between the hyperplane and the data points on both sides are equal. In this case, the interval can be expressed as $2/\|w\|$. Therefore, the problem of constructing an optimal hyperplane is transformed into the optimization problem $\min \|w\|^2/2$ under the above constraints.

In the case of linear separability, the optimal weight vector w^* and optimal bias b^* can be solved by using the Lagrange function and dual method, and then the optimal classification hyperplane $(w^* \cdot x) + b^* = 0$ can be obtained.

In the case of linear inseparability, the main idea of the SVM algorithm is to map the input vector x to a high-dimensional eigenvector space and to construct the optimal classification plane in the eigenspace. x is mapped from the input space R^n to the eigenspace H using map ϕ, and we obtain

$$x \longrightarrow \phi(x) = (\phi_1(x), \phi_2(x), \ldots, \phi_l(x))^T. \tag{2}$$

Replacing x with the eigenvector $\phi(x)$, similar to the linear separability case, we can obtain the optimal weight vector w^{**} and optimal bias b^{**} and then the optimal classification hyperplane $(w^{**} \cdot \phi(x)) + b^{**} = 0$ in the high-dimensional feature vector space can be determined.

2.3. RF. The random forest (RF) method [20] is a representative integrated learning method. The basic idea of an integrated learning method is to train multiple weak classifiers and to combine them into one strong classifier.

With the bootstrap method, k different new training sets are constructed from the original training set, and each new training set is used to train a decision tree separately. Utilizing different training sets can increase the difference between classification models, improving the generalization ability of the combined classification model. After k rounds of training, a classification model sequence $\{h_1(X), h_2(X), \ldots, h_k(X)\}$ is obtained and used to form a multiclassification model system, which utilizes simple majority voting to obtain the final classification results. The final classification result is

$$H(x) = \arg\max_{Y} \sum_{i=1}^{k} I(h_i(x) = Y), \tag{3}$$

where $H(x)$ is the combinatorial classification model, h_i is a single decision tree classification model, Y is the output category, and $I(\cdot)$ is a schematic function.

2.4. AdaBoost. The adaptive boosting (AdaBoost) [21] approach is an improvement on the boosting algorithm, which is an integrated learning method. In contrast to the RF method, in which the weak classifiers are independent of each other, the weak classifiers in the AdaBoost approach are not independent of each other, and the sample weight of the new classifier is adjusted according to the results of the preorder classifier.

Given a training set $D = \{(x_i, y_i) | \ i = 1, 2, \ldots, m, \quad x \in R^n, \quad y \in \{\pm 1\}\}$, in the first round of training, the training set is assigned to be evenly distributed (that is, each training sample has the same weight), and a weak classifier is trained on it. Then, the training set is updated according to the training results (the weights of misclassified samples are increased), and the new training set is used for training again. After k rounds of training, the classification model sequence $\{h_1(X), h_2(X), \ldots, h_k(X)\}$ is finally obtained. Each classifier has a certain weight, and the final classification model is obtained by voting with the weights:

$$H(x) = \arg\max_{Y} \sum_{i=1}^{k} \alpha_i I(h_i(x) = Y), \tag{4}$$

where α_i is the weight of classifier $h_i(X)$.

2.5. MLP. The multilayer perceptron (MLP) [22], developed from the single-layer perceptron that cannot solve nonlinear classification problems, is the basic theory of neural network/deep learning. The basic purpose of a perceptron is to simulate biological neurons. This method sums each input in a weighted manner and converts the inputs using an activation function to obtain output results, which can be used for classification problems.

The single-layer perceptron is the simplest type of neural network, with only input and output layers, which means that only one round of weighted summation and transformation is implemented. The MLP includes hidden layers as well, thereby realizing multiple rounds and making it more powerful. Figure 1 shows the basic structure of the

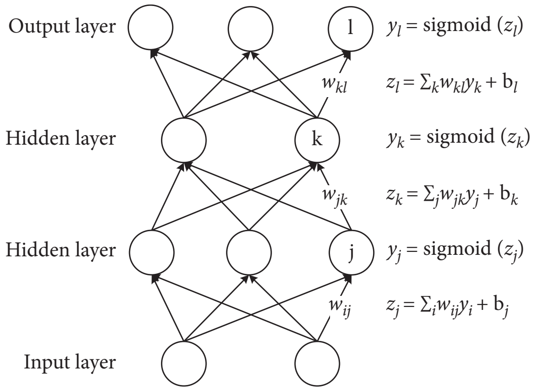

Figure 1: MLP.

MLP. Each layer contains a number of nodes. ω_{ij}, ω_{jk}, and ω_{kl} are the connection weights between layers; b_j, b_k, and b_l are the biases of the layers; and z_j, z_k, and z_l are the sums of the inputs and biases. Further, y_j, y_k, and y_l are the outputs of the sigmoid function (z_j, z_k, and z_l are the inputs), which is the predicted category in the classification problem. The weights of the connections are the parameters to be trained, which are adjusted by backpropagation.

3. Data Acquisition and Feature Extraction

This section describes the Bitcoin transaction dataset and the extraction of the transaction characteristics.

3.1. Dataset Description. We used the Bitcoin transaction dataset published by Harrigan (https://anonymity-in-bitcoinblogspotcom/2011/09/). The dataset depicts Bitcoin transaction data using three directed graphs: a transaction graph (including 1,019,486 transaction vertices and 1,558,854 transaction edges), user graph (including 926,615 user vertices and 1,961,636 user edges), and public key graph (including 1,253,054 public key vertices and 3,491,341 public key edges). We mainly used the transaction graph for our experiments. The vertices of the transaction graph are the Bitcoin transaction hash, and the edges are the Bitcoin flows between transactions. If one output of a transaction is used as input for another transaction, then there is a directed edge between the two transactions and the weight of the directed edge is the amount of Bitcoin transferred.

To use a supervised machine learning algorithm, anomalous transaction data must be flagged. To this end, we searched for reports of Bitcoin theft related to the dataset of Harrigan from the Bitcoin forum website Bitcointalk (https://bitcointalkorg/indexphp?topic=576337). By viewing the reports and querying Blockchain Explorer, we correlated the transactions involved in theft with the transaction dataset of Harrigan. We finally marked 568 transactions in the dataset as theft transactions. Bitcoin theft transaction detection can be regarded as binary classification with unbalanced data, where the amount of abnormal data is usually much smaller than the amount of normal data. Directly using machine learning to categorize unbalanced data will be less effective because of insufficient learning of minority classes. Consequently, oversampling and undersampling are typically used to process training data to improve the learning effect. To address the unevenness of the dataset and to ensure the efficiency of the algorithm, we undersampled the nontheft transactions (i.e., we randomly selected 10,000 of the 1,018,918 nontheft transactions) and merged them with the 568 theft transactions as our experimental data.

3.2. Feature Extraction. Bitcoin theft refers to a hacker stealing the private key of a Bitcoin user and transferring Bitcoin from the address of the user through the Internet or by other means. Our analyses and assumptions regarding Bitcoin theft (as well as the feature extraction process) are as follows. In the case of Bitcoin theft, after the hacker steals Bitcoin from the victim, the hacker usually transfers the Bitcoin out as soon as possible to avoid the account from being frozen. Therefore, the waiting time interval from a transaction's creation to the spending of the maximum output of the transaction is taken as the time interval feature. Because the transaction adjacent to a theft transaction is likely to be a theft transaction as well, we extracted the mean and maximum of the time interval feature of the input transactions and the output transactions as neighbor features. In theft activity, the hacker usually transfers Bitcoin efficiently by using a relatively large total amount and centralized transaction outputs. Therefore, we extracted the total transaction amount and variance of the transaction output as features. In addition, we extracted the number of decimal places of the maximum output, input transactions quantity, and output transactions quantity as the digit, indegree, and outdegree features, respectively. Table 1 summarizes the features we extracted and their definitions.

4. Experimental Results and Discussion

4.1. Experimental Results. We extracted nine features from the collected historical Bitcoin transaction data, standardized the features, and divided the training and test sets in a ratio of 7 : 3. We then experimented with eight machine learning methods from the sklearn library of Python. The abnormal data proportion parameter of the three unsupervised methods was set as 0.05, and the rest of the parameters were set as default values. For the five supervised methods, parameters are all set as default values.

Unlike the commonly used metric error rate in classification problems, due to the particularity of anomaly detection problems, we used the recall, precision, and $F1$ score as the main evaluation metrics in the experiment. These metrics are defined as follows (the meanings of TP, TN, FP, and FN are as usually defined in machine learning theory):

Recall: TP/(TP+FN)

Precision: TP/(TP + FP)

$F1$: (recall × precision)/(recall + precision)

To verify the effectiveness of this method, three groups of experiments were designed. First, the effectiveness of the supervised methods compared with the unsupervised methods was verified using three features. Then, the effectiveness of the supervised methods compared with the

Table 1: Extracted features.

Feature name	Definition
Time interval	Wait time for the maximum output of a transaction to be spent
Variance of transaction output	Variance of all the output amounts of the transaction
Number of decimal digits	Number of digits after the decimal point of the maximum transaction output amount
Mean time interval of input transactions	Average of the time intervals of all input transactions
Mean time intervals of output transactions	Average of the time intervals of all output transactions
Maximum time interval of output transactions	Maximum value of the time intervals of all output transactions
Outdegree	Number of output transactions
Indegree	Number of input transactions
Total amount	Total value of a transaction

unsupervised methods was verified using nine features, and the effectiveness of designed nine features was illustrated. Finally, the effectiveness of oversampling the training set to improve the detection results was verified. We refer to the above three groups of experiments as experiments A, B, and C, respectively, and Table 2 summaries the results of them.

In the first set of experiments, we utilized three unsupervised methods—the LOF, Mahalanobis distance-based method, and OCSVM algorithm—as well as five supervised methods—the KNN, RF, AdaBoost, SVM, and MLP approach—on three features—the indegree, outdegree, and total amount—which are described in the literatures [8, 9]. The recall and precision are close to zero for all three unsupervised methods, with LOF being slightly superior. As LOF is suitable for detecting relatively isolated points locally and the other two methods are suitable for detecting points at the edge of the whole, this indicates that abnormal data points tend to be distributed within the whole data set. Among the supervised methods, the KNN, RF, and AdaBoost approaches achieved recall and precision values of approximately 60%, which are significantly improved compared to those of the unsupervised methods. The highest $F1$ value in this set of experiments is that of the KNN algorithm, which is 60.4%.

In the second set of experiments, we utilized the eight methods from the first set of experiments with nine features (expanded features). Among the unsupervised methods, the recall and precision of the LOF approach and the Mahalanobis distance-based method are improved, but both indicators are still less than 20%. As to the supervised methods, the recall and precision of the KNN algorithm are both over 80%, and these indicators exceed 90% for the RF and AdaBoost methods. Meanwhile, the recall and precision of SVM and MLP are still zero. The highest $F1$ value in this set of experiments is that of the RF approach, which is 95.2%.

In the third set of experiments, we performed the synthetic minority oversampling technique (SMOTE) [23] on the minority class in the training set and utilized all five supervised methods on nine features. The recall of the KNN approach is more than 90% and those of the RF and AdaBoost methods are more than 95%, but the precision of these three methods decreased significantly. The recall of SVM and MLP is more than 90%, but the precision is less than 60%, which is significantly lower than the other three methods. The highest $F1$ value in this set of experiments is that of the RF approach, which is 95.9%.

On the left-side figure of Figure 2, we show the distribution of the test set itself. The red points represent the abnormal transactions, and the blue points represent the normal transactions. As most transactions are in the lower left corner, we enlarged the local area in the lower left, as we did in Figures 3 and 4. In the left and right enlarged graphs of Figure 3, we present the classification results of the test set using the KNN method, RF method, and AdaBoost method before and after equalization, respectively. It can be seen that the classification results of the three methods before and after equalization are quite close to the distribution of the test set itself. For KNN and AdaBoost, equalization may make more normal points be classified as anomalies. In the left and right enlarged graphs of Figure 4, we show the classification results of the test set using the SVM method and MLP method before and after equalization, respectively. Compared with the distribution of the test set itself, we can see that the effect of the two methods has been significantly improved after equalization.

4.2. Discussion. In the past, research on the detection of Bitcoin theft was relatively rare, and the detection effect of Bitcoin theft in the existing literature was not ideal. The methods they use are mostly unsupervised methods, and the extracted features are relatively simple. In view of the fact that some Bitcoin thefts have been reported, we used this information to label the data and accordingly utilized five supervised methods. The recall and precision of the three supervised methods, KNN, RF, and AdaBoost, are significantly improved compared to those of the unsupervised methods, when both three and nine features are used. This finding explains to some extent the advantages of using supervised algorithms. The other two supervised methods, SVM and MLP, have zero recall with both three and nine features. There are two possible reasons for this behavior: (1) severe dataset imbalance can skew the

Table 2: Experimental results corresponding to various settings.

			Acc (%)	Recall (%)	Pre (%)	*F*1 (%)
A	Unsupervised	LOF	89.5	0.0	0.0	—
		OCSVM	82.8	0.5	0.2	0.3
		MDB	89.6	0.5	0.6	0.6
	Supervised	KNN	95.5	62.2	58.7	60.4
		SVM	94.5	0.0	0.0	—
		RF	95.5	56.3	59.1	57.7
		AdaBoost	95.7	52.9	62.3	57.2
		MLP	94.5	0.0	0.0	—
B	Unsupervised	LOF	89.6	0.5	0.6	0.6
		OCSVM	89.7	0.0	0.0	—
		MDB	89.5	0.0	0.0	—
	Supervised	KNN	98.5	84.8	87.4	86.1
		SVM	94.5	0.0	0.0	—
		RF	99.4	92.4	98.1	95.2
		AdaBoost	99.3	93.6	94.7	94.1
		MLP	94.4	0.0	0.0	—
C	Supervised	KNN	97. 6	92. 4	72. 2	81. 1
		SVM	50. 1	90. 1	9. 0	16. 3
		RF	99. 5	95. 9	95. 9	95. 9
		AdaBoost	98. 7	97. 0	82. 6	89. 3
		MLP	95. 9	93. 0	57. 9	71. 4

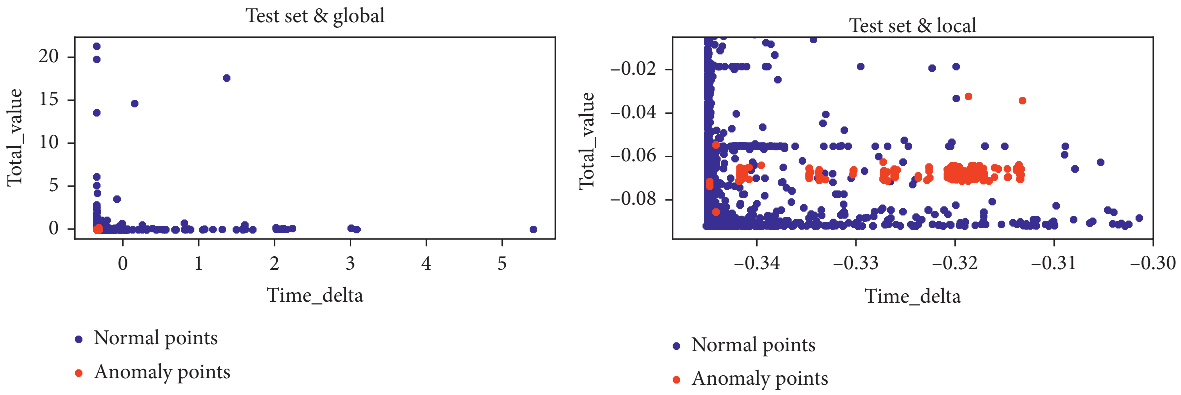

Figure 2: Distribution of test set (global and local).

decision boundaries more toward the minority class and (2) the feature we designed still has a low ability to distinguish between abnormal data and nonabnormal data.

To distinguish between theft and nontheft transactions more effectively, we expanded upon the features used in previous studies [8, 9] to obtain nine features. According to the above results, two of the three unsupervised detection methods exhibit improved detection capabilities with our extended features, and the improvement effect of the OCSVM method is more obvious. Meanwhile, the recall and precision of three of the five supervised methods are significantly improved by using our extended nine features compared to those resulting from using the original three features. This finding shows the effectiveness of the features we designed for supervised methods; that is, it demonstrates that these features can highlight theft transactions to a certain extent.

As dataset imbalance will lead to poor classification of minority classes, we adopted the SMOTE method to equalize the training set. In Table 3, we highlight the changes of metrics of five supervised methods before and after equalization. B, C, and RC represent the second group of experiments, the third group of experiments, and the rate of change (in percentage terms) of the three main metrics, respectively. Equalization significantly improved the recall of all five supervised methods, especially that of the SVM and MLP methods, which increased from zero to more than 90%. Simultaneously, there is a certain decline in

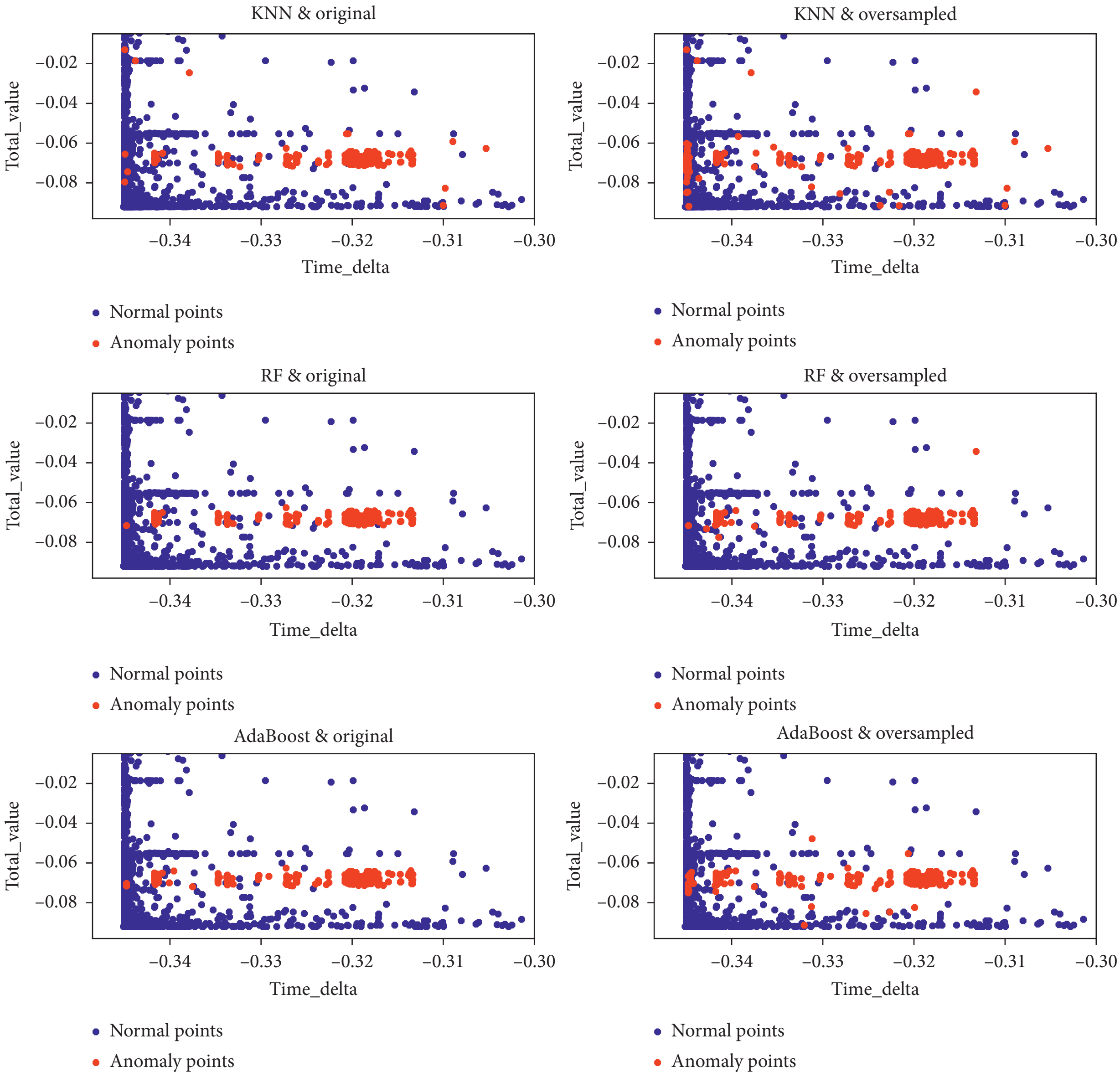

FIGURE 3: Classification results of test set before and after training set oversampling (KNN, RF, and AdaBoost).

precision, less than 20%, for KNN, RF, and AdaBoost. The best performing method is the RF approach. Both before and after oversampling, $F1$ is the highest for the RF approach among the five methods, and this value is improved by equalization. The negative effect of the decrease in precision is smaller than the positive effect of the increase in recall. As shown in Figure 5, after oversampling, FN decreases by 6 and FP increases by 4. Because the cost of FN is greater than that of FP, the total cost will drop. In terms of cost and $F1$, equalization is effective for the RF approach. From the perspective of recall, equalization is effective for all five supervised methods.

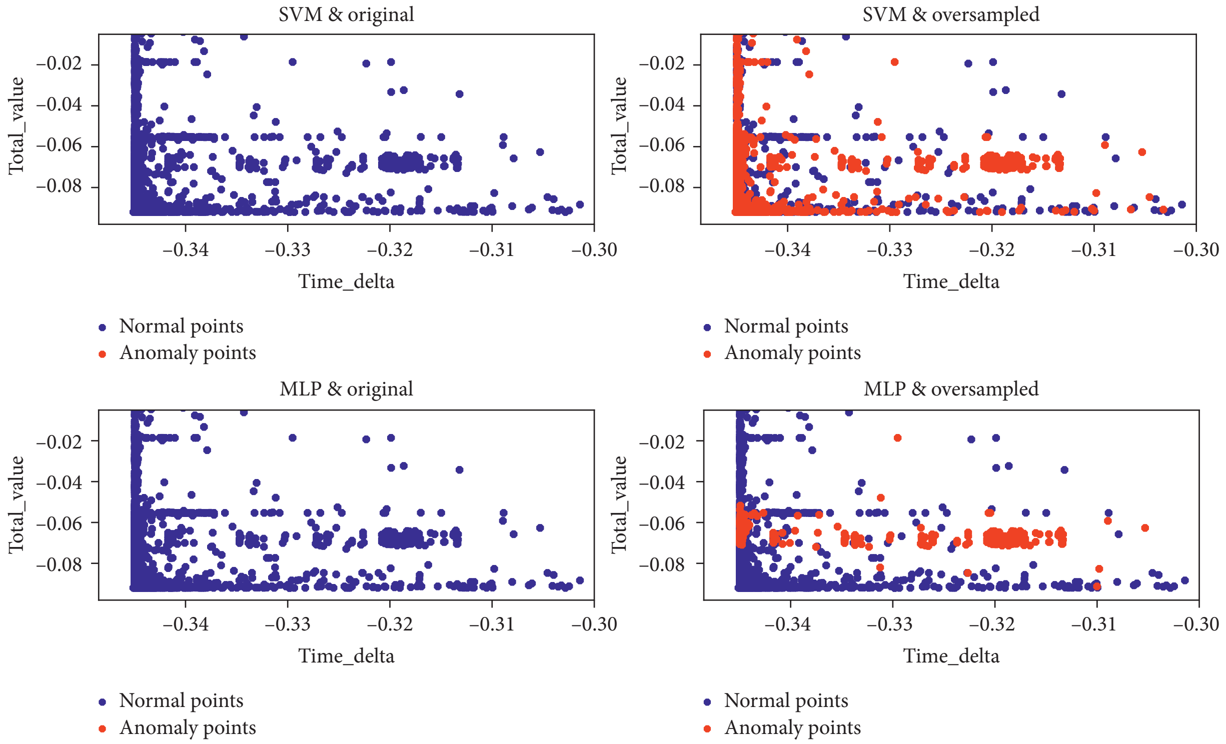

Figure 4: Classification results of test set before and after training set oversampling (SVM and MLP).

Table 3: Effectiveness of test set classification before and after training set oversampling.

	Recall (%)			Precision (%)			*F1* (%)		
	B	C	RC	B	C	RC	B	C	RC
KNN	84.8	92.4	8.9	87.4	72.2	−17.3	86.1	81.1	−5.8
SVM	0.0	90.1	—	0.0	9.0	—	0.0	16.3	—
RF	92.4	95.9	3.7	98.1	95.9	−2.2	95.2	95.9	0.7
AdaBoost	93.6	97.0	3.6	94.7	82.6	−12.7	94.1	89.3	−5.1
MLP	0.0	93.0	—	0.0	57.9	—	0.0	71.4	—

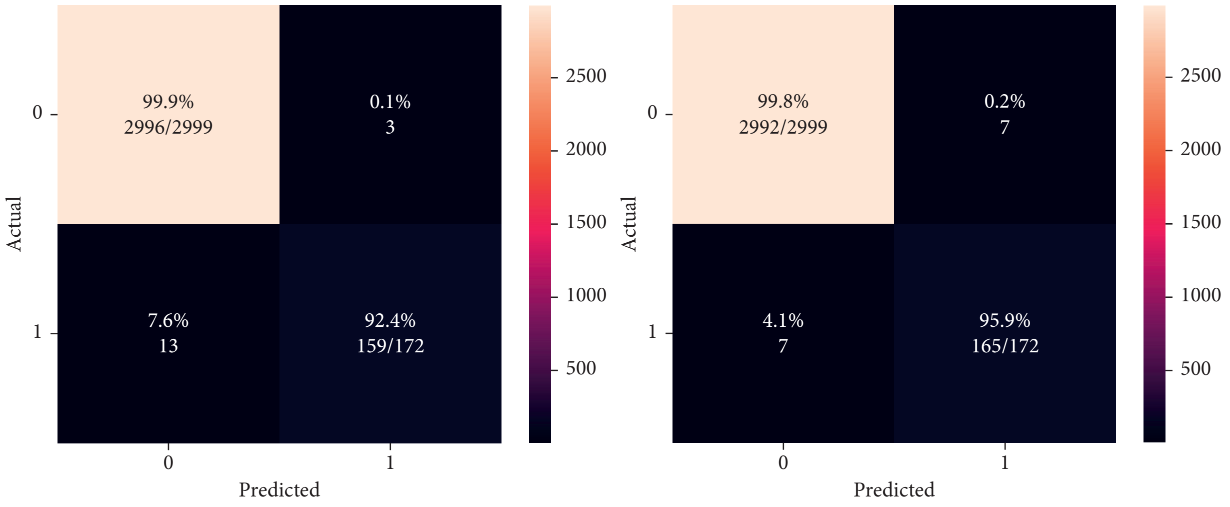

Figure 5: Confusion matrices of the RF method before and after training set oversampling.

5. Conclusions and Future Studies

In this study, we focused on the detection of Bitcoin theft transactions using supervised machine learning methods. We collected historical Bitcoin transaction data and extracted nine features based on the characteristics of theft transactions. We used five supervised methods—the KNN, SVM, RF, AdaBoost, and MLP methods—to classify the features. The experimental results showed that the RF, AdaBoost, and KNN algorithms have better classification abilities than the other approaches, with *F*1 values of 95.2%, 94.1%, and 86.1%, respectively. Further, we performed oversampling to equalize the unbalanced training set. The experiments showed that the recall was further improved by equalization. Among the investigated approaches, the RF method exhibited the best classification performance, with its F1 value reaching 95.9%.

Our next research directions include two aspects: (1) extracting more targeted features for theft transactions and combining multiple machine learning algorithms to improve the detection results and (2) using homomorphic encryption algorithms to encrypt features to achieve theft transaction detection with privacy protection.

Data Availability

All data included in this study are available upon request from the corresponding author.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

References

[1] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: architecture, consensus, and future trends," in *Proceedings of the 5th IEEE International Congress on Big Data(Big Data Congress 2017)*, pp. 557–564, Boston, MA, USA, June 2017.

[2] K.-K. R. Choo, Z. Yan, and W. Meng, "Editorial: blockchain in industrial IoT applications: security and privacy advances, challenges, and opportunities," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4119–4121, 2020.

[3] T. McGhin, K.-K. R. Choo, C. Z. Liu, and D. He, "Blockchain in healthcare applications: research challenges and opportunities," *Journal of Network and Computer Applications*, vol. 135, pp. 62–75, 2019.

[4] G. Karame, "On the security and scalability of bitcoin's blockchain," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security(CCS 2016)*, pp. 1861-1862, Vienna, Austria, October 2016.

[5] J. Wang, X. Gu, W. Liu, and A. K. Sangaiah, "An empower Hamilton loop based data collection algorithm with mobile agent for WSNs," *Human-centric Computing and Information Sciences*, vol. 18, no. 9, pp. 1–14, 2019.

[6] Y. Chen, J. Wang, R. Xia, Q. Zhang, Z. Cao, and K. Yang, "The visual object tracking algorithm research based on adaptive combination kernel," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 12, pp. 4855–4867, 2019.

[7] B. Yin and X. Wei, "Communication-efficient data aggregation tree construction for complex queries in IoT applications," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3352–3363, 2019.

[8] T. Pham and S. Lee, "Anomaly detection in bitcoin network using unsupervised learning methods," 2016, https://arxiv.org/abs/1611.03941.

[9] T. Pham and S. Lee, "Anomaly detection in the bitcoin system—a network perspective," 2016, https://arxiv.org/abs/1611.03942.

[10] K. Toyoda, T. Ohtsuki, and P. T. Mathiopoulos, "Identification of high yielding investment programs in Bitcoin via transactions pattern analysis," in *Proceedings of the 17th IEEE Global Communications Conference(GLOBECOM 2017)*, pp. 1–6, Singapore, December 2017.

[11] M. Vasek and T. Moore, "Analyzing the bitcoin Ponzi scheme ecosystem," in *Proceedings of the 22nd International Conference On Financial Cryptography And Data Security(FC 2018)*, pp. 101–112, Nieuwpoort, Curaçao, February 2018.

[12] W. Chen, Z. Zheng, J. Cui et al., "Detecting Ponzi schemes on Ethereum: towards healthier blockchain technology," in *Proceedings of the 27th World Wide Web Conference(WWW 2018)*, pp. 1409–1418, Lyon, France, April 2018.

[13] C. F. Torres and M. Steichen, "The art of the scam: demystifying honeypots in Ethereum smart contracts," in *Proceedings of the 28th USENIX Security Symposium(USENIX Security 2019)*, pp. 1591–1607, Santa Clara, CA, USA, August 2019.

[14] W. Chen, J. Wu, Z. Zheng et al., "Market manipulation of bitcoin: evidence from mining the Mt. gox transaction network," in *Proceedings of the 38th IEEE Conference on Computer Communications(INFOCOM 2019)*, pp. 964–972, Paris, France, April 2019.

[15] W. Chen, Y. J. Xu, Z. Zheng et al., "Detecting "pump & dump schemes" on cryptocurrency market using an improved Apriori algorithm," in *Proceedings Of the 13th IEEE International Conference On Service-Oriented System Engineering (SOSE 2019)*, pp. 293–2935, San Francisco, CA, USA, April 2019.

[16] L. Yang, X. Dong, S. Xing et al., "An abnormal transaction detection mechanim on bitco*in*," in *Proceedings of the 4th International Conference on Networking and Network Applications (NaNA 2019)*, pp. 452–457, IEEE, Xian, China, October 2019.

[17] M. Bartoletti, S. Carta, T. Cimoli, and R. Saia, "Dissecting Ponzi schemes on Ethereum: identification, analysis, and impact," *Future Generation Computer Systems*, vol. 102, pp. 259–277, 2020.

[18] S. Garcia, J. Derrac, J. R. Cano, and F. Herrera, "Prototype selection for nearest neighbor classification: taxonomy and empirical study," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 3, pp. 417–435, 2012.

[19] D. Zhang, G. Yang, F. Li, J. Wang, and A. K. Sangaiah, "Detecting seam carved images using uniform local binary patterns," *Multimedia Tools and Applications*, vol. 79, no. 13, pp. 8415–8430, 2020.

[20] J. Zhang, X. Jin, J. Sun, J. Wang, and A. K. Sangaiah, "Spatial and semantic convolutional features for robust visual object tracking," *Multimedia Tools and Applications*, vol. 79, no. 21, pp. 15095–15115, 2020.

[21] H. Liu, H.-Q. Tian, Y.-F. Li, and L. Zhang, "Comparison of four Adaboost algorithm based artificial neural networks in wind speed predictions," *Energy Conversion and Management*, vol. 92, pp. 67–81, 2015.

[22] J. Tang, C. Deng, and G. B. Huang, “Extreme learning machine for multilayer perceptron,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 4, pp. 809–821, 2015.

[23] P. Jeatrakul, K. W. Wong, and C. C. Fung, “Classification of imbalanced data by combining the complementary neural network and SMOTE algorithm,” in *Proceedings of the 17th International Conference On Neural Information Processing(ICONIP 2010)*, pp. 152–159, Sydney, Australia, November 2010.