How to Find a Bitcoin Mixer: A Dual Ensemble Model for Bitcoin Mixing Service Detection

Chang Xu, Ruting Xiong, Xiaodong Shen, Liehuang Zhu, *Senior Member, IEEE*, and Xiaoming Zhang

***Abstract*—Bitcoin is the first decentralized peer-to-peer cryptocurrency that has gained popularity by providing users with transaction anonymity. With the development of Bitcoin and the higher privacy requirements of users, mixing services have emerged to enhance Bitcoin anonymity by obfuscating the flow of funds. However, they are also widely used for illegal activities due to its strong anonymity, especially for money laundering. Therefore, detecting mixing services has great significance for Bitcoin anti-money laundering. In this article, we propose a novel detection scheme to identify the addresses belonging to Bitcoin mixing services. Specifically, we first construct the Bitcoin mixing data set, which summarizes a total of 26 features to describe the transaction behavior of addresses. Next, we design a new classification model, called the Dual Ensemble Classification Model. The model combines the advantages of multiple models based on different algorithms and obtains better classification performance. In order to detect more complex mixing patterns, we also extract transaction subgraphs from the established Bitcoin address-transaction network. The subgraphs are then classified using a kernel-based graph classification method, which is embedded in the model. Comprehensive experiments on three data sets demonstrate the effectiveness of our scheme, and the proposed model has a detection accuracy of 99.84% for the Bitcoin mixing service.**

***Index Terms*—Bitcoin, ensemble learning, kernel-based graphical classification, mixing service.**

I. Introduction

BITCOIN is a decentralized public ledger system that enables users to use addresses to participate in bitcoin transactions without revealing their actual identity. The anonymity of Bitcoin is that users can hold any number of addresses, while an address can only correspond to one user. As a result, it is difficult to establish an exact match between the address and the user, which fosters illegal activity. For example, illegal activities, such as underground markets, ransomware attacks, money laundering [1], [2], and Ponzi schemes [3] all use Bitcoin in their transactions. And mixing services are extensively employed in these criminal activities to aid in money laundering. A study discovered that the Silk Road,[1] which offers illegal items and services, commonly uses mixing services [4]. Furthermore, on 8 May 2019, around 7000 bitcoins were stolen from Binance [5], and most Bitcoins were later transferred to the prominent mixing service provider Chipmixer [6].

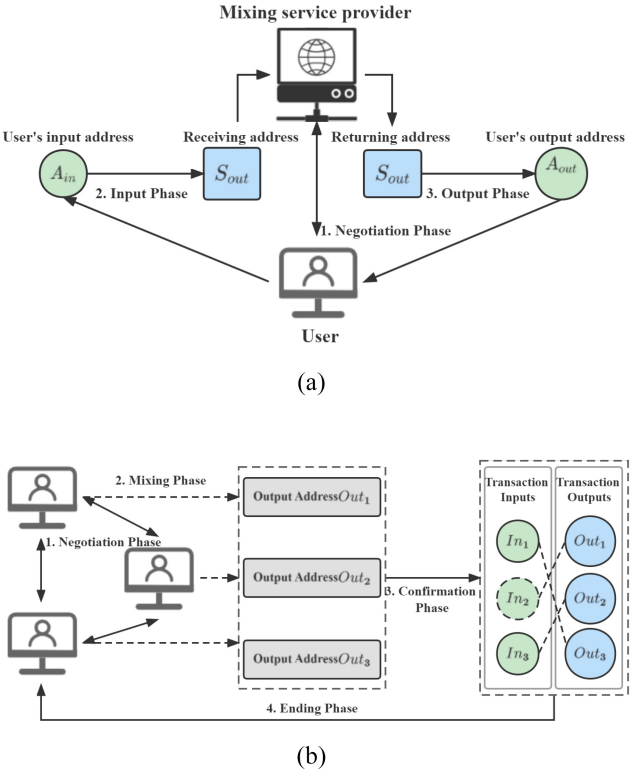

Fig. 1. Mixing technology schematic. (a) Centralized mixing technology. (b) Decentralized mixing technology.

Fig. 1 describes the process of the centralized mixing service and the decentralized mixing service, respectively. The centralized mixing technology requires the centralized mixing service providers to help users with mixing operations. In the decentralized mixing technology, all users who participate in the mixing process spontaneously conduct currency mixing transactions according to the agreement. Overall, mixing services are performed by pooling funds from multiple inputs over a long period of random time and then returning them to destination addresses. Since all funds are grouped

Manuscript received 12 December 2022; revised 26 March 2023; accepted 28 April 2023. Date of publication 11 May 2023; date of current version 25 September 2023. This work was supported by the National Key Research and Development Program of China under Grant 2021YFB2700500 and Grant 2021YFB2700502. *(Corresponding author: Chang Xu.)*

Chang Xu, Ruting Xiong, Xiaodong Shen, and Liehuang Zhu are with the School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing 100081, China (e-mail: xuchang@bit.edu.cn; rutingx@bit.edu.cn; shenxiaodong@bit.edu.cn; liehuangz@bit.edu.cn).

Xiaoming Zhang is with the Key Laboratory of Aerospace Network Security, Ministry of Industry and Information Technology, School of Cyber Science and Technology, Beihang University, Beijing 100191, China (e-mail: yolixs@buaa.edu.cn).

Digital Object Identifier 10.1109/JIOT.2023.3275158

[1]https://en.wikipedia.org/wiki/Silk_Road_(marketplace)

and distributed at random times, it is difficult to keep track of funds flows.[2]

The widespread use of mixing services makes it difficult to track money laundering activity on Bitcoin. Unfortunately, the detection of mixing services has not received enough attention in Bitcoin anti-money laundering efforts. Wu et al. [7] revealed some key transaction behaviors that can distinguish the mixing service addresses, and designed an effective mixing service detection model. However, the model requires prior knowledge of mixing rules, resulting it incapable of detecting new complicated mixing patterns. Wu et al. [8] presented a heuristic-based algorithm to identify mixing transactions, but there is some bias in evaluating the effectiveness of the method due to the lack of ground-truth data. Balthasar and Hernandez-Castro [1] only focused on analyzing mixing services themselves, but did not apply the method to detect mixing services.

In this work, we focus on the issue of Bitcoin mixing service detection. To solve the problem of lacking ground-truth data, we crawl the labeled addresses belonging to mixing services and regular service from WalletExplorer.com.[3] We get the real Bitcoin transaction data related to these addresses from Blockchain.com.[4] Then, we build a Bitcoin mixing data set with more than 10 000 instances. Additionally, for detecting more complex mixing patterns, we model the Bitcoin transaction data as the Bitcoin address-transaction network and extract transaction subgraphs centered on each address. After labeling the subgraphs, we apply the kernel-based graph classification technology to identify transaction subgraphs with central addresses belonging to mixing services. Consequently, the graph structure characteristics of Bitcoin transaction network can be fully utilized to describe the transaction behavior of address nodes.

As illustrated in Fig. 2, the proposed Bitcoin mixing service detection scheme proposed in this article consists of four phases.

1) *Data Collection:* We first obtain actual Bitcoin transaction data via the blockchain data interface provided on Blockchain.com. Then, we merge the data with the tagging data captured from WalletExplorer.com to construct a Bitcoin mixing data set.
2) *Visualization and Feature Extraction:* We use Neo4j, a graphical database, to store and analyze Bitcoin transaction data. Then, we extract statistical features from transaction records at different levels.
3) *Extract Transaction Subgraphs:* Subgraphs are extracted from the Bitcoin address-transaction network and labeled to form a subgraph data set.
4) *Model Training and Testing:* The Bitcoin mixing data set and the transaction subgraph data set are divided to train and test our model. Furthermore, we use two relevant data sets from other research to validate the effectiveness of the model.

In summary, this article makes the following main contributions.

1) We construct and visualize the Bitcoin address-transaction network, and summarize 26 statistical features from the original Bitcoin transaction records to build the Bitcoin mixing data set. In addition, we extract subgraphs from the network to form a transaction subgraph data set.
2) We design a new dual ensemble classification model, which contains six component models. Models that have performed well in previous work are integrated into our model for better performance. And, we innovatively add a kernel-based graph classification model into our model to detect more complex mixing patterns. To the best of our knowledge, this article is the first to apply ensemble learning and kernel-based graph classification technology to the Bitcoin mixing service detection problem.
3) We evaluate the designed model on three different data sets, including the Bitcoin mixing data set, the Ethereum account data set [9], and the Elliptic data set [10]. Experimental results show that our model outperforms both the baseline models and the adaptive stacked extreme gradient boosting (ASXGB) model [11] on different data sets.

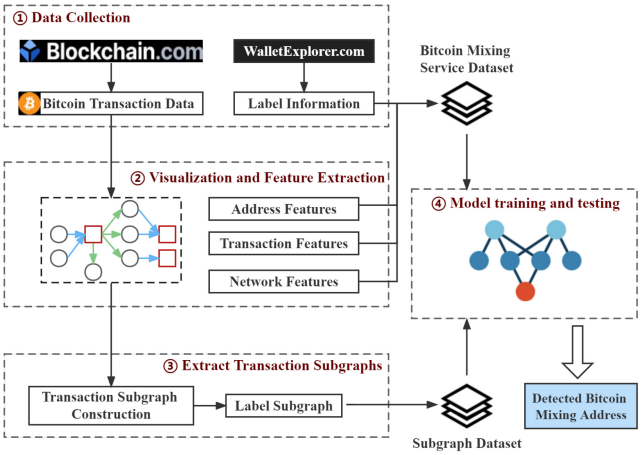

Fig. 2. Overview of the Bitcoin mixing detection framework.

The remainder of this article is organized as follows. Section II describes our data collection process and introduces the two additional data sets used. Section III gives a detailed description of our model. Then, we present the experimental results in Section IV. Finally, we provide related work in Section V and conclude this article in Section VI.

II. Data Set Description

The following part of this section moves on to describe in detail the three data sets used in this article. We first introduce the construction process of Bitcoin mixing data set and describe the features we selected from three levels. And, we also give a brief introduction about the Ethereum account data set and the Elliptic data set.

A. Bitcoin Mixing Data Set

1) Data Collection: WalletExplorer.com is an intelligent Bitcoin blockchain explorer with two features: 1) different addresses that are considered to belong to the same user are

[2]https://en.wikipedia.org/wiki/Cryptocurrency_tumbler
[3]https://www.walletexplorer.com
[4]https://www.blockchain.com

TABLE I
STATISTICS OF BITCOIN MIXING SERVICE DATA SET

Service	No. of address	No. of Tx.s	Percentage
BitcoinFog	5107	44849	49.05%
Binance	1965	4973	18.87%
BitcoinWallet	1378	94400	13.23%
CoinPayments	1962	60854	18.84%

merged together and 2) each wallet is given a name to distinguish the different entities. Blockchain.com is a provider of crypto products that provides various blockchain data APIs, and everyone can easily access transaction data on the blockchain through the APIs on it.

Using some public media and reports as our information sources, we select one representative mixing service and three regular services as our research objects, which are listed as follows.

1) BitcionFog[5] is an online Bitcoin wallet that can only be accessed through the Tor network. It offers the functionality to send transactions anonymously, and hide the source of cryptocurrency. It is reportedly notorious for providing money laundering services to criminals. Over the past ten years, the site has moved more than 1.2 million Bitcoins, valued at around $335 million.
2) Binance[6] is a global cryptocurrency exchange that provides a trading platform for over 100 cryptocurrencies. And, it is considered the largest cryptocurrency exchange in the world in terms of trading volume since the beginning of 2018.
3) BitcoinWallet[7] is the official Bitcoin wallet, and since 2014, nearly one million users have relied on Bitcoinwallet.com as their official Bitcoin wallet.
4) CoinPayments[8] is another popular Bitcoin wallet, which has advantages such as industry-low processing fees of just 0.5% and real-time global payments.

We crawl the data of addresses belonging to these four services from WalletExplorer.com, and label all addresses belonging to Biannce, BitcoinWallet, and CoinPayments as regular addresses and those associated with BitcoinFog as mixing addresses. In order to get more detailed information, we use the APIs on Blockchain.com to capture the data of transactions in which each address is involved. Table I shows the summary statistics for the data collection, including the number of addresses and transactions of each service, as well as the percentage of labeled mixing and regular addresses among all addresses.

2) Feature Characterization: The purpose of the Bitcoin mixing service is to obfuscate the flow of funds and increase the anonymity of cryptocurrencies. Therefore, the addresses and transactions associated with them differ from those related to regular service. In the following, we provide the features extracted from three levels in turn.

[5]https://bitcoinfog.co/
[6]https://www.binance.com/en
[7]https://bitcoinwallet.com/
[8]https://www.coinpayments.net/

TABLE II
ADDRESS FEATURES AND ITS INTERPRETATION

Feature	Interpretation
balance	The balance of address
in_trans_num	The number of transactions which the address acts as the input address
out_trans_num	The number of transactions which the address acts as the output address
max_trans_per_day	The maximum number of daily transactions of each address
avg_val	The average of the values transformed from/to each address
std_val	The standard deviation of the values transformed from/to each address
in_gini	The Gini coefficient of the values transformed to the address
out_gini	The Gini coefficient of the values transformed from the address

TABLE III
TRANSACTION FEATURES AND ITS INTERPRETATION

Feature	Interpretation
mean_vin_sz	Average number of transaction inputs
mean_vout_sz	Average number of transaction outputs
std_vin_sz	Standard deviation of the number of transaction inputs
std_vout_sz	Standard deviation of the number of transaction outputs
mean_size	Average size of transactions
std_size	Standard deviation of transactions size
mean_fee	Average fee of transactions
std_fee	Standard deviation of transactions fee
mean_time_diff	Average time between transactions
std_time_diff	Standard deviation of time between transactions

Address Features: In many cases, the state and activeness of an address can indicate which category the address belongs to. Accordingly, we apply address features to describe the state and activeness of them. For example, addresses associated with Bitcoin exchanges typically have a higher trade frequency for a wide range of businesses, while the trade frequency of many ordinary users is relatively much lower. The extracted address features are detailed in Table II.

Transaction Features: Each address is involved in numerous transactions that contain a large amount of data. Consequently, it is possible to measure the transaction behavior of addresses during the mixing process by making full use of the transaction characteristics. Table III summarizes the transaction features.

Fig. 3 is a heat-map showing the Pearson correlation between address features and transaction features. The Pearson correlation coefficient measures the degree of linear correlation between two variables. When the Pearson correlation coefficient value is smaller than 0.3, the two variables are considered weakly correlated. Note that there are 66 correlation coefficients in Fig. 3, and most of which are smaller than 0.3. Hence, training models with data sets made up of these features can produce better results, because there is less redundant data and each feature can provide some unique information in the model's learning process.

Network Features: The raw Bitcoin transaction records can be considered as a transaction network due to Bitcoin's peer-to-peer structure. As shown in Fig. 4, we construct a Bitcoin address-transaction graph $G = (V, E, T)$ using the collected address and transaction data, where V is the node set and E is

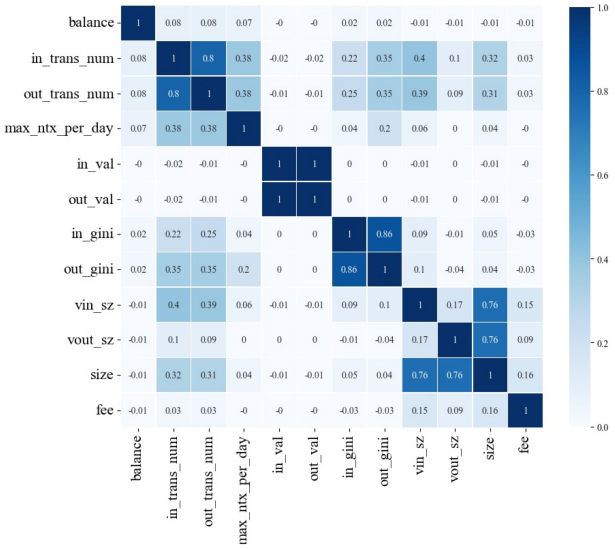

Fig. 3. Pearson correlation coefficient of features.

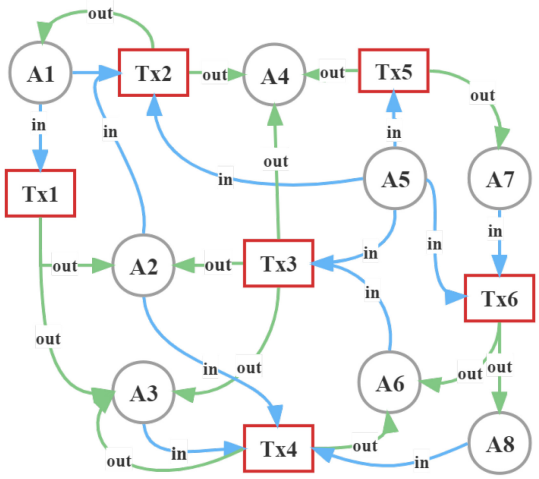

Fig. 4. Schematic of the Bitcoin address-transaction network.

a set of edges. $T = \{\text{Address}, \text{Transaction}\}$ represents the set of node types. For any node $v \in V$, its node type $\phi(v) \in T$. And for each edge $e_{u,v} \in E$ between node u and v, $\phi(u) \neq \phi(v)$, and either $\phi(u) = \{\text{Address}\}$ or $\phi(v) = \{\text{Address}\}$. This means that each edge represents the process of Bitcoin transfer between addresses and transactions.

Neo4j is an NOSQL graphical database that stores structured data on the Web. We use it to store and visualize Bitcoin address and transaction data gathered from the four services. Fig. 5 provides the results of the visualization, where the red nodes represent the transaction, the gray nodes represent the address, the green edges denote the output of the transaction, and the blue edges denote the input of the transaction. In Fig. 5(a) and (b), the red transaction nodes are organized into two distinct circular formations, herein referred to as the *inner layer* and the *outer layer*. Notably, the inner layer comprises the transactions with a relatively small number of inputs and outputs, which are termed as *lightweight transactions*. Conversely, the outer layer is characterized by the transactions with a single input but multiple outputs, where the number

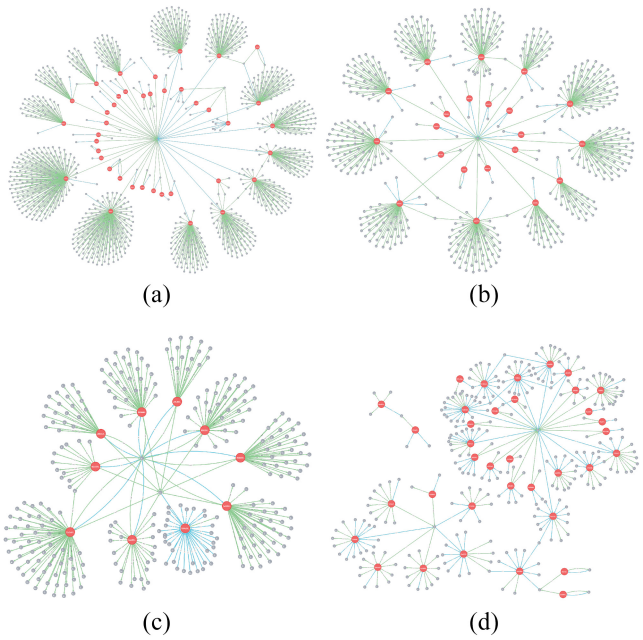

Fig. 5. Bitcoin transaction data visualization of four services. (a) Binance. (b) BitcoinWallet. (c) CoinPayments. (d) BitcoinFog.

of outputs substantially exceeds the number of inputs, thus termed as *output-type transactions*. We designate the transactions with a substantially larger amount of input relative to the output as *input-type transactions*.

From the observation of these address-transaction networks, we summarize the following findings.

1) *Finding 1:* The mean value of address participation (MAP) of transactions in Binance, BitcoinWallet, and Coinpayments is higher than those in BitcoinFog. MAP is calculated as follows:

$$\text{MAP} = \frac{\text{all\_in} + \text{all\_out}}{\text{all\_txs}}$$

where all_in is the sum of input addresses, all_out is the sum of output addresses, and all_*txs* is the total number of transactions.

2) *Finding 2:* In Binance, BitcoinWallet, and Coinpayments, transaction types are clearly characterized. There are two types of transactions in Binance and BitcoinWallet, i.e., the output-type transactions on the outer layer and the lightweight transactions on the inner layer. Coinpayments contains one input-type transaction and the rest are output-type transactions. However, the transaction types of BitcoinFog are diverse and lack such distinguishing characteristics.
3) *Finding 3:* There are fully connected addresses (addresses that are associated with all transactions in the visualization section) in Binance, BitcoinWallet, and Coinpayments, while there is no such address in BitcoinFog.

Through the above analysis, we extract the following network features.

1) MAP.
2) Number of output-type transactions (the difference between the number of output addresses and the number of input addresses is greater than a threshold value).

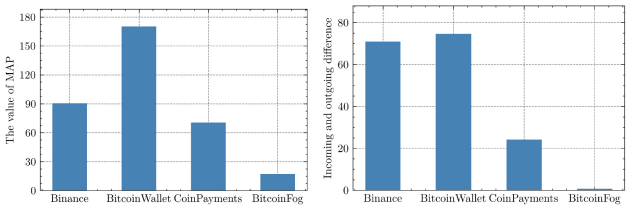

Fig. 6. Comparison of network features for different services. (a) Value of MAP. (b) Difference between outgoing and incoming.

3) Number of input-type transactions (the difference between the number of input addresses and the number of output addresses is greater than threshold value).
4) Number of central addresses (the number of transactions directly associated with this address is greater than a threshold value).
5) Average out-degree of transaction nodes.
6) Average in-degree of transaction nodes.

Fig. 6 compares two network features of the four services. As can be seen, the MAP value of the mixing service BitcoinFog is lower than those of the other three. The result is consistent with the fact that the transaction network of BitcoinFog is sparser in the visualization. We also compare the difference between outgoing and incoming of transaction nodes of the four services. There is a significant difference of this feature between BitcoinFog and the other three services. This indicates that the number of inputs and outputs of transactions in mixing service are essentially equal, which is due to the working principle of the mixing service. As mixing services usually pool funds from multiple sources and send them to different addresses to maintain the dispersion and blurriness of the flow of funds, there are no divergent transactions with few inputs and many outputs, or clustered transactions with many inputs and few outputs, which would expose the user's address.

B. Ethereum Account Data Set

The second data set used in this work is the Ethereum account data set from the study by Farrugia et al. [9]. There are 2179 illicit accounts and 2502 licit accounts in this data set. The Ether community flagged illegal accounts for illegal behavior, which includes: 1) scam lotteries; 2) impersonating other users; 3) fake initial coin offerings (ICO); 4) phishing; 5) mirroring websites; 6) attempting to imitate other contract addresses providing tokens; and 7) Ponzi schemes. Based on historical transactions, a total of 42 features were extracted for each account in the data set.

C. Elliptic Data Set

Elliptic data set is the largest tagged transaction data set publicly accessible in all cryptocurrencies. Each transaction in the elliptic data is represented as a vertex in a transactional graph, with directed edges signifying the flow of money. The data set consists of 203 796 transactions with a total of 234 355 directed edges. Among all of these transactions, 4545 were flagged as illegal, 42 019 as legal, and the rest were left unmarked. Every transaction contains 166 attributes, which are divided into two groups: 1) local and 2) aggregated features. In this work, we do not consider the unlabeled transactions in the data set and only use the labeled 46 564 transactions. Table IV summarizes the basic information of the three data sets.

TABLE IV
BASIC INFORMATION OF DATA SETS

Dataset	No. of features	No. of samples	Positive to negative sample ratio
Bitcoin mixing service dataset	24	10412	$\approx 1:1$
Ethereum account dataset	42	4682	$\approx 1:1$
The Elliptic dataset	165	46565	$1:10$

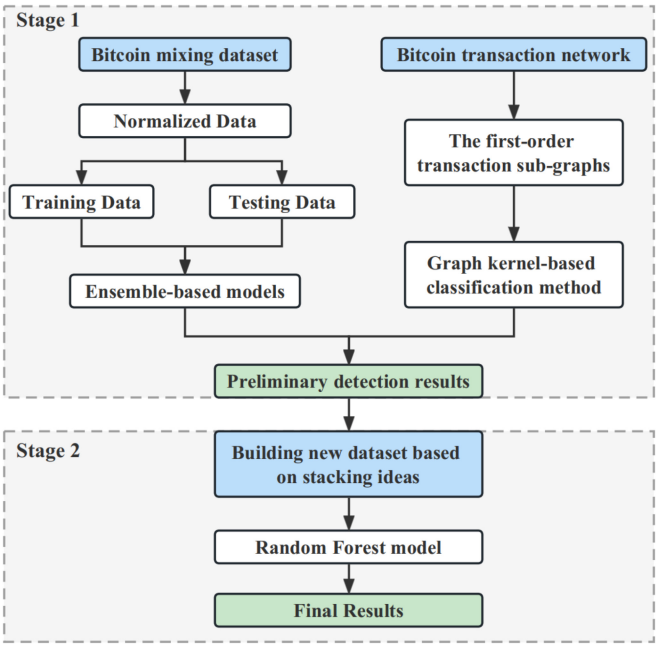

Fig. 7. Dual ensemble classification model architecture.

III. METHODOLOGY

Although various machine learning models have been applied to blockchain anomaly transaction detection, most of them are classical algorithms [3], [12], [13] and few improved models have been proposed. We present a dual ensemble model to detect Bitcoin mixing services, taking into account the complexity and uniqueness of Bitcoin transaction data. Fig. 7 depicts the architecture of the suggested model. First, we employ a range of ensemble classifiers and graph kernel-based classification algorithms for preliminary detection. Due to the possibility of model overfitting in ensemble classifiers, we do not focus on ajusting the parameters of these models, but utilize a new classifier to integrate the outputs of these models to achieve a more accurate end result. Each component of the proposed model is described in detail next.

A. Preliminary Detection Using Ensemble Models

Ensemble learning refers to the method of building a series of learners and using certain rules to integrate their result to obtain a better learning effect than a single learner [14].

The existing ensemble learning approaches can be classified into two types based on the generation mechanism of individual learners, namely, Boosting and Bagging. Considering that to obtain a decent ensemble model, individual learners must be "excellent but different," we choose four types of ensemble classifiers that have been demonstrated to perform well in detecting abnormal blockchain transactions for preliminary detection [15], [16]. Among them, AdaBoost, GBDT [17], eXtreme gradient boosting (XGBoost) [18], and LightGBM [19] are all ensemble classifiers based on the idea of Boosting. We first divide the Bitcoin mixing service data set D into a training set D_{train} and a test set D_{test}

$$D = (x_i, y_i)\big(|D| = n, x_i \in X \subseteq R^{n\times M}, y_i \in \{0, 1\}\big)$$

where x_i denotes the ith instance, y_i denotes its label, and M is the feature dimension of the instance. Then, we train the model in the following four steps.

1) To train the base classifier, a training subset is chosen at random from the training set D_{train} using no-return sampling.
2) Based on the samples misclassified by the previous classifier, the weights of each sample in the training subset are adjusted to train a new base classifier.
3) Steps 1 and 2 are repeated M times, where M is the number of base classifiers.
4) According to the performance of each weak classifier, the weight of its contribution to the overall result is then decided.

Specifically, the final classifier of the AdaBoost model is

$$G(x) = \text{sign}\left[\sum_{m=1}^{M} \alpha_m G_m(x)\right]$$

where $\alpha_m = (1/2)\log[(1 - e_m/e_m)]$, and e_m is the classification error rate of $G_m(x)$ on the training subset. The final classifier of the GBDT model is

$$F_M(x) = F_0(x) + \sum_{m=1}^{M}\sum_{j=1}^{J_m} c_{m,j} I\big(x \in R_{m,j}\big)$$

where $c_{m,j}$ is the best fit value and $F_0(x)$ is the initial weak classifier. The XGBoost model optimizes the GBDT model by adding regular terms to the objective function to prevent overfitting of the model, and its objective function equation is as follows:

$$\text{obj}(\theta) = \sum_{i}^{n} l\big(y_i, \hat{y}_i\big) + \sum_{k=1}^{K} \Omega(f_k).$$

In addition, it can automatically parallelize compute processes using the CPU's multithreads to increase accuracy based on GBDT. We also employed the LighGBM model, which is a more effective tree-enhanced model than GBDT and XGBoost in terms of computational performance and memory consumption.

B. Graph Classification Using the Graph Kernel-Based Classification Method

As mentioned in Section II, we represent Bitcoin transaction data as a Bitcoin transaction network $G = (V, E, T)$. To fully utilize the topological information of nodes in the network to describe the transaction behavior of addresses, we extract the first-order transaction subgraphs from the entire Bitcoin transaction network, with each address node as the center, and then label the subgraph with the central address. The particular procedure is as follows.

Extracting First-Order Transaction Subgraphs: For any node $u \in V$, and $\phi(u) = \{\text{Address}\}$, we consider a directed weighted graph $g_u = (V, E, B)$. On g_u, $V = \{u, \tilde{V}\}$, $\phi(\tilde{V}) = \{\text{Transaction}\}$ and for any node $v \in \tilde{V}$, it is a neighboring node of u. For each $e_{u,v} \in E$ between u and its neighboring node v, we have $\phi(u) = \{\text{Address}\}$ and $\phi(v) = \{\text{Transaction}\}$, which means the edge $e \in E$ represent the coin transfer from or to address node u.

Labeling First-Order Transaction Subgraphs: For every subgraph g_u that we extract from the whole Bitcoin transaction network, we give it a label, let $\text{label}(g_u) = \text{lable}(u)$. That is, the label of the transaction subgraph is the same as the label of its central address node. The reason is the transaction subgraph of the address does not contain the transaction information of other addresses. If address u belongs to the mixing service, the transaction subgraph related to u is different from that related to the regular address. Therefore, the label of the address can be used as the label of its transaction subgraph. Finally, we obtain a subgraph data set $D_G = (g_i, y_i)(|D| = n, g_i \in G, y_i \in \{0, 1\})$, where g_i denotes a subgraph object in the Bitcoin transaction network centered on address instance x_i.

Classification of First-Order Transaction Subgraphs: After the above two steps are executed, we transform the Bitcoin mixing services detection problem into a graph classification problem. Given a set of graphs, the goal of graph classification is to learn the mapping relationship between the graphs and the corresponding category labels, and to predict the category labels of unknown graphs [20]. Graph kernel methods combine the representational power of graphs with the differentiation power of kernel methods differentiation capability of graphs [21], which can complete graph classification tasks based on graph similarity computation.

In this article, we use the graph kernel method for graph classification and use the shortest path kernel as the kernel function. Because the size of the transaction subgraph is relatively small, the advantages of the graph kernel method can be taken full use of without consuming excessive computational resources. The shortest path kernel method first transforms the original graph into a shortest path graph, where the set of nodes in the shortest path graph is the same as the set of nodes in the original graph, except that the edges in the shortest path graph are labeled with the shortest path lengths in the two nodes. Let G_1 and G_2 be two input graphs, and the shortest path graphs be S_1 and S_2, respectively. Then, define the shortest path graph kernel based on S_1 and S_2

$$k_{\text{shortest}-\text{path}}(S_1, S_2) = \sum_{e_1\in E_1}\sum_{e_2\in E_2} k^{(1)}_{\text{walk}}(e_1, e_2)$$

where $k^{(1)}_{\text{walk}}$ is a semi-positive definite kernel on the edge, which can be defined as a Dirac kernel function, i.e., its value is 1 when e_1 and e_2 have the same length and 0 otherwise.

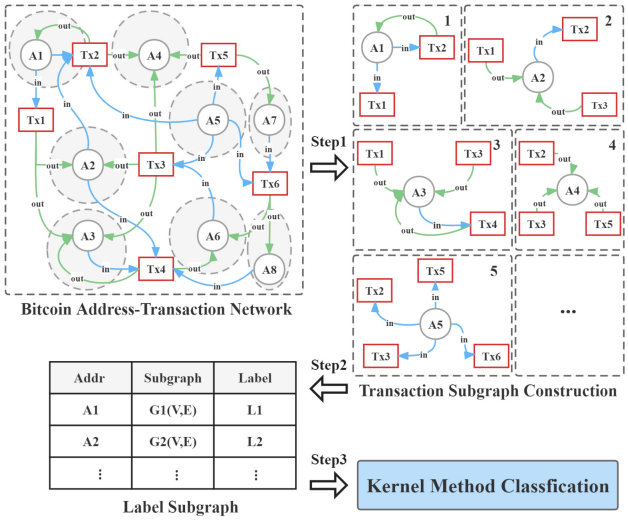

Fig. 8. Schematic of the process by using graph classification method to detect Bitcoin mixing services.

Therefore, it can be seen that the more shortest paths of the same length, the higher the similarity of the graph. The whole process of detecting bitcoin mixing services using the graph kernel method is shown in Fig. 8.

C. Combining Models Using Stacking Ideas

We performed preliminary detection of Bitcoin mixing services using four distinct types of ensemble classifiers. And taking the network structure of bitcoin transactions into account, we also employ graph kernel classification algorithms for the detection task. To further improve the accuracy of the detection results, we combine these models using a stacking integration strategy [22] to take advantage of their strengths. On the one hand, the stacking strategy allows for the combination of different types of models, and their combination can lead to a more accurate prediction. On the other hand, stacking can help to reduce overfitting. In our scheme, multiple base models with different algorithms and hyperparameters are trained on the same data set. Thus, our model is less likely to memorize noise in the training data and can capture the underlying patterns that generalize well to new data.

Integration of Models: The first-level learners are four classic ensemble learning models: 1) AdaBoost; 2) GBDT; 3) XGBoost; and 4) LightGBM, and a shortest path kernel-based graph classification model. Let $f_i = \{0, 1\}$ denote the result produced by the first-level learners. Based on the results of the first-level learners, we construct a new data set $D_{\text{new}} = (f_i, y_i)(|D_{\text{new}}| = n, y_i \in \{0, 1\})$ to train the second-level learner. We choose the Random Forest [23] model as the second-level learner because of its simplicity and good predictive performance. The output of the second-level learner is the final classification result of the model. The entire classification process of our model is described in Algorithm 1.

Algorithm 1 Pseudocode of the Dual Ensemble Classification Model

Require: The Bitcoin mixing dataset: $D = \{X, Y\}$; The subgraph dataset: $D_G = \{G, Y\}$;
Ensure: The class $\hat{y}$ to which instance x belongs
1: AdaBoost_Model.fit(X)
2: $f_{i1} \leftarrow$ AdaBoost_Model.predict(x_i)
3: GBDT_Model.fit(X)
4: $f_{i2} \leftarrow$ GBDT_Model.predict(x_i)
5: XGBoost_Model.fit(X)
6: $f_{i3} \leftarrow$ XGBoost_Model.predict(x_i)
7: LightGBM_Model.fit(X)
8: $f_{i4} \leftarrow$ LightGBM_Model.predict(x_i)
9: SPKernelGraphModel.fit(G)
10: $f_{i5} \leftarrow$SPKernelGraphModel.predict(g_i)
11: $D_{new} \leftarrow$ ConstructNewDataset(f_i, y_i)
12: RF_Model.fit(F)
13: $\hat{y}_i \leftarrow$RF_Model.predict(f_i)
14: **return** $\hat{y}$

IV. Experiment

In this section, we first outline the experimental setting and evaluation metrics. Afterwards, we introduce the results of experiments which are conducted to evaluate the proposed scheme, along with an analysis of the findings.

A. Experimental Settings

The benchmarks are the basic learners in the suggested model. AdaBoost, GBDT, XGBoost, and LightGBM are implemented using the scikit learning package [24], and the graph classification model based on the shortest path kernel is implemented using the GraKel learning package [25].

By eliminating missing values, there are 9740 remaining instances in the Bitcoin mixing service data set, whereas the Ethereum accounts data set contains only 4681 total instances. Since the number of valid instances in these two data sets is less than 10 000, we evaluate each model on them with ten-fold cross-validation for obtaining more effective information from the data. In the Elliptic data set, there are up to 46 565 available instances. When conducting experiments, we choose 70% of the Elliptic data set at random as the training set and 30% as the test set.

B. Evaluation Metrics

We evaluate the performance of our model using several metrics that are commonly used to evaluate the performance of classification models, which are briefly described in the following.

1) Accuracy measures the proportion of correctly classified samples in all samples, which is calculated as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN}.$$

2) Precision measures the proportion of true positive samples among the predicted positive samples, it is formulated as follows:

$$\text{Precision} = \frac{TP}{TP + FP}.$$

3) Recall measures the proportion of positive samples in all samples that are correctly predicted, the formula of it is in the following:

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

where TP and FP represent the number of items correctly and incorrectly labeled as belonging to the positive class, respectively. FN is the number of items incorrectly labeled as belonging to the negative class. TN is the number of items correctly labeled as belonging to the negative class.

4) F1-score, the harmonic mean of precision and recall, is used to measure the performance of the classifier when the sample categories are not balanced. It is formula is as follows:

$$F_1 = \frac{2(\text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})}.$$

5) The receiver operating characteristic (ROC) curve is used to measure the discriminative ability of the classifier system when the classification threshold changes. AUC refers to the area under the ROC curve, which is generally between 0.5 and 1.

6) The Kolmogorov–Smirnov (KS) measures the difference between the cumulative distributions of good and bad samples. The greater the cumulative difference between good and bad samples, the larger the KS value, then the better the model's ability to differentiate. The KS is calculated as below

$$\text{KS} = \max(\text{TPR} - \text{FPR})$$

where True Positive Rate ($TPR = TP/(TP + FN)$), namely, *Sensitivity*, represents the ratio of actual positive instances to all positive instances in the positive class predicted by the classifier. And, Negative Positive Rate ($FPR = FP/(FP+TN)$), namely, *1-Specificity*, represents the proportion of actual negative instances in the positive class predicted by the classifier to all negative instances.

C. Method Comparison

We initially compare the performance of the proposed model with the baseline models on the Bitcoin mixing data set in order to validate our model's capacity of identifying mixing services. Fig. 9 provides the results, and the specific data of the experimental findings are provided in Table V. It is apparent that our model outperforms all basic models on various evaluation metrics. Specifically, the suggested model outperforms all single models by around 9% in accuracy and precision and by about 8% in F1-score and recall. The causes of excellent performance of the suggested model thanks to its smoothing nature and the ability to highlight each base model on its best performance cases. Moreover, the diversity of primary models is a key factor in enhancing its effectiveness.

Then, we compare our model with the ASXGB model from Vassallo et al. [11]. Vassallo claimed ASXGB is an adaptation of XGBoost, and successfully reduced the impact of concept drift while further improving recall at a transaction level in anti-money laundering in cryptocurrencies. Ethereum accounts data set and Elliptic data set were utilized to evaluate ASXGB in their work. To demonstrate efficiency of our model on classification tasks, we first compare it with ASXGB using these two data sets. The experimental results are shown in Fig. 10, where Dataset1 represents the Ethereum account data set, Dataset2 represents the Elliptic data set, and Dataset3

TABLE V
PERFORMANCE COMPARISON OF THE MODELS ON BITCOIN MIXING DATA SET

Model	Accuracy	Precision	F1-score	Recall
AdaBoost	90.26%	90.43%	91.10%	91.81%
GBDT	89.97%	90.27%	91.07%	91.92%
XGBoost	90.29%	90.60%	91.30%	92.06%
LightGBM	90.45%	90.69%	91.32%	91.98%
Graph model	89.47%	96.95%	86.68%	78.44%
Random Forest	90.36%	90.76%	91.35%	92.06%
Proposed Model	99.50%	99.84%	99.55%	99.28%

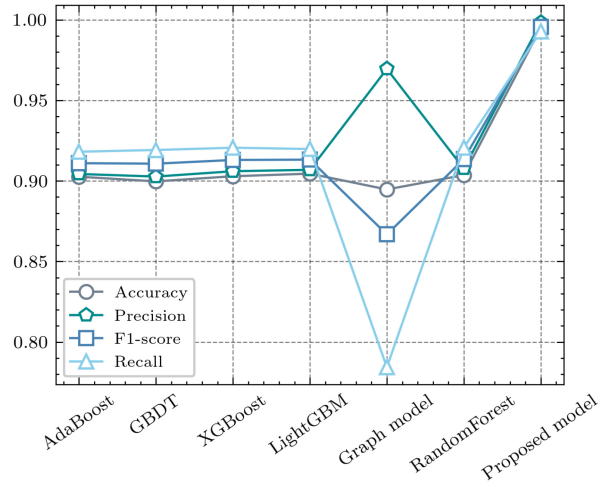

Fig. 9. Performance comparison of different models on Bitcoin mixing data set.

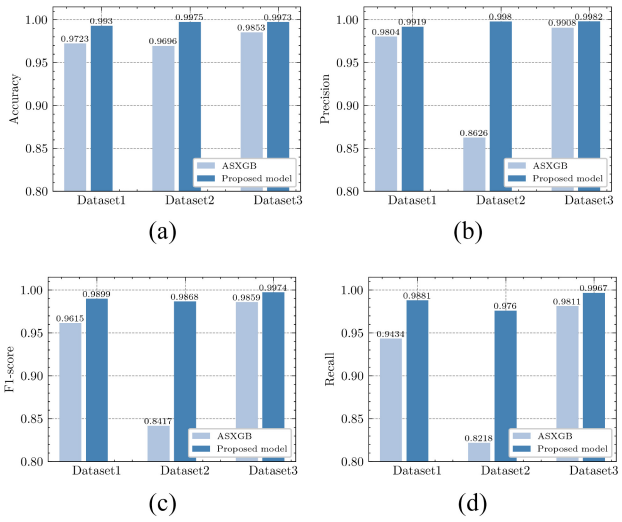

Fig. 10. Performance comparison of the proposed model with ASXGB model on different data sets. (a) Accuracy. (b) Precision. (c) F1-score. (d) Recall.

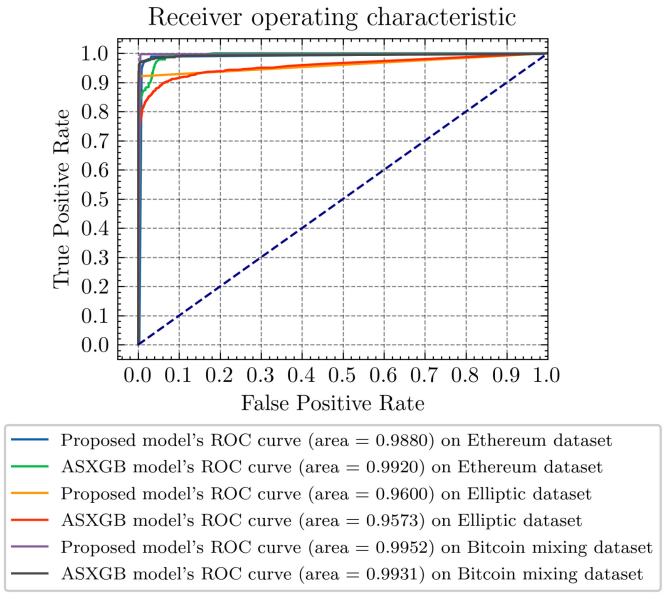

Fig. 11. ROC curves of ASXGB and proposed model on different data sets.

represents our Bitcoin mixing data set. It can be seen that our model outperforms ASXGB under all data sets. Moreover, compared to the performance on other data sets, there is a significant reduction in ASXGB's performance on Elliptic data set.

Additionally, comparison experiments between the proposed model and ASXGB model were conducted using the Bitcoin mixing data set. Results are shown in Fig. 10, our model is more effective on detecting Bitcoin mixing services. First, the graph kernel-based classification model can leverage the topological characteristics of each address in the Bitcoin transaction network, which improves the performance of the scheme. Second, the stacking ensemble strategy holds the advantages of all base models and can reduce overfitting. Therefore, our scheme is superior to ASXGB.

Fig. 11 is the ROC curves of ASXGB and the suggested model on three data sets. All of the ROC curves in the figure are toward the upper left corner, demonstrating the effectiveness of both models. With the exception of the Ethereum accounts data set, our model's AUC values on the other two data sets are higher than ASXGB's. The KS curves of ASXGB and our model on three data sets are shown in Fig. 12. As mentioned in Section IV-B, the KS value measures the maximum vertical distance between the TPR and FPR, i.e., the maximum distance between the two curves. Larger KS values indicate better the discrimination ability of the model. The KS curve graphs indicate that the proposed model outperforms the ASXGB model, as evidenced by the higher KS value achieved by the former.

D. Efficiency Evaluation

We evaluate the efficiency of the proposed scheme on a computer with 16-GB memory and windows11 operating system. The experiment was repeated five times, and the average value of the running time is shown in Table VI. The results of our experiments indicate that the training time for

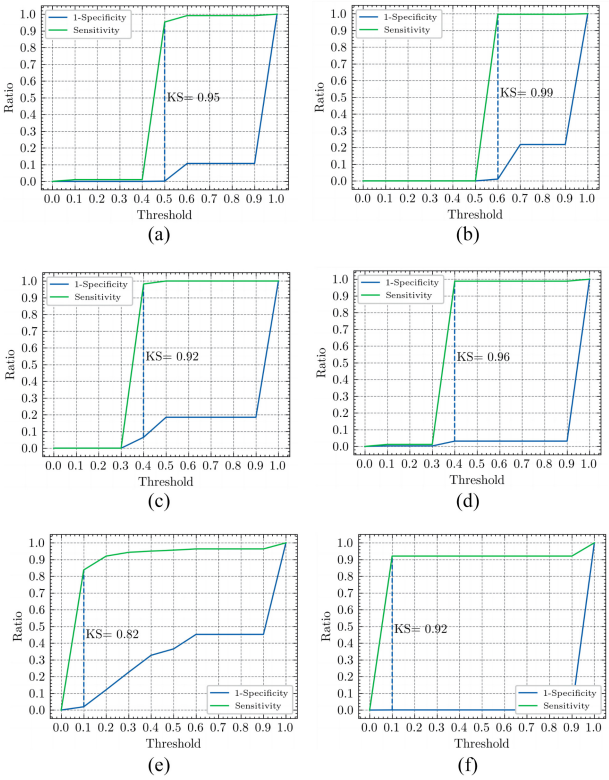

Fig. 12. KS curves of ASXGB and proposed model on different data sets. (a) KS curve of ASXGB on Bitcoin mixing data set. (b) KS curve of proposed model on Bitcoin mixing data set. (c) KS curve of ASXGB on Ethereum account data set. (d) KS curve of proposed model on Ethereum account data set. (e) KS curve of ASXGB on Elliptic data set. (f) KS curve of proposed model on Elliptic data set.

TABLE VI
RUNNING TIME OF EACH PART IN THE PROPOSED SCHEME

	Model	Running Time
The first level	AdaBoost	4.92s
	GBDT	21.29s
	XGBoost	2.03s
	LightGBM	0.78s
	Graph model	95.7s
	Construct new dataset	1.24ss
The second level	RandomForest	6.58s

different models varied significantly. LightGBM only took 0.78 s to generate results, while the Graph model took the longest time 95.7 s. Among the other models, XGBoost performed the best, completing its training process in only 2.03 s, followed by AdaBoost (4.92 s), RandomForest (6.58 s), and GBDT (21.29 s). The first-level models are independent of each other and can process data in parallel. Therefore, the total running time of this scheme is $95.7 + 1.24 + 6.58 = 103.52$ s. Although this is longer than the running time of a single model, the reduction in efficiency is still within an acceptable range in consideration of the 9% increase in accuracy.

V. RELATED WORK

The anonymity of Bitcoin stems from the fact that the user controls the address for transactions and it is not possible to establish an exact inverse mapping of addresses to users. Since the creation of Bitcoin in 2009, research

and attacks on anonymity have not stopped, scholars have proposed many ways to exploit the transaction data to undermine Bitcoin's anonymity. Reid and Harrigan [26] found that Bitcoin addresses can be linked to real-life users using external information, such as user registration details and voluntarily disclosed public keys. Meiklejohn et al. [27] proposed two heuristic address clustering algorithms to identify addresses belonging to the same entity in bitcoin: 1) the multi-input heuristic and 2) the change addresses heuristic. Liu et al. [28] used machine learning techniques to transform the address clustering problem into a binary classification problem that incorporates the features of two bitcoin addresses.

As a result, cryptocurrency mixing services have emerged to enhance Bitcoin's anonymity. The basic idea of mixing services is to confuse the relationship between sender and receiver to maintain the anonymity of the relationship. With the widespread use of mixing services, there are some works to study their working principle and to reveal the mixing process. Möser et al. [2] conducted the first empirical study, which systematically elaborated and evaluated three mixing services in the Bitcoin ecosystem. Balthasar and Hernandez-Castro [1] analyzed selected services using tools provided by Chainalysis[9] and discovered serious security flaws in these services. Pakki et al. [29] provided a more recent survey on mixing services in Bitcoin, in which the author provides a table of mixing services with nine trusted services. Wu et al. [8] proposed a generic model to systematically analyze state-of-the-art mixing services. These efforts simply examined mixing services and make no recommendations for detecting them.

Using mixing services to improve transaction anonymity helps protect users' privacy to a larger level, but anonymity also gives convenience for illegal and criminal activities. The main illegal use of mixing services is to assist in money laundering. In view of the problem of cryptocurrency money laundering using mixing services, Wu et al. [7] provided a feature-based network analysis framework and proposed the concept of attributed temporal heterogeneous motifs (ATH motifs) to detect addresses belonging to mixing services to assist anti-money laundering in Bitcoin. Their detection method is based on some known mixing rules, thus, it is difficult to detect unknown mixing patterns. Wu et al. [8] proposed a method to identify mixing transactions that leverage the obfuscating mechanism, and provided a case study of tracing the money flow of stolen Bitcoins. However, the mixing mechanism is not the same for different mixing services, so these methods are often difficult to generalize.

With the development of machine learning, there has been a lot of work using machine learning techniques to detect illegal activities based on mixing services. Weber et al. [10] proposed a study on detecting Bitcoin money laundering transactions using network characteristics and node embedding. Vassallo et al. [11] provided the ASXGB model to detect illegal activities on cryptocurrencies. Alarab et al. [30] suggested adding a linear layer to the graph convolutional network to improve the performance of the detection model. Hu et al. [31] recommended and evaluated a set of classifiers based on four types of graph features: 1) immediate neighbors; 2) curated features; 3) deepwalk embeddings; and 4) node2vec embeddings to classify money laundering and regular transactions. Most of these schemes using machine learning models focus on feature engineering rather than model improvement, so the models are not innovative enough.

[9]https://www.chainalysis.com/

VI. Conclusion

In this work, we designed a new Bitcoin mixing service detection model, namely, the dual ensemble classification model. The principle of stacking is used to combine the different classification models that have done well in prior work, which can play the advantages of each model and obtain a better performance of the combined model. AdaBoost, GBDT, XGBoost, and LightGBM are four of the six component learners that were used, and they are all classic models based on the ensemble learning concept.

To mine the information in the Bitcoin transaction data more thoroughly, we additionally modeled it as the Bitcoin address-transaction network, extracted the transaction subgraph from it, and built a subgraph data set. The suggested model's classification performance is enhanced by the addition of the kernel-based graph classification approach, which fully utilizes the address's graph structure information. Finally, experiments on Bitcoin mixing data set demonstrated that the proposed model can effectively identify the addresses participating in the coin mixing process. Once the addresses belonging to the mixing services are identified, the flow of funds can be traced through the transaction chain, providing help for the anti-money laundering. Besides, our scheme is experimented on the Elliptic data set, which is widely used to evaluate models for detecting cryptocurrency money laundering transactions. The experimental results indicate that our model can detect the Bitcoin laundering transactions with a high accuracy of 99.75%.

A limitation of this study is that due to the limited label information available on WalletExplorer.com, the Bitcoin mixing data set did not cover all addresses of different mixing services. In future work, we will collect label information from more sources, such as interacting with well-known mixing services to obtain richer data.

References

[1] T. D. Balthasar and J. Hernandez-Castro, "An analysis of bitcoin laundry services," in *Proc. Nordic Conf. Secure IT Syst.*, 2017, pp. 297–312.
[2] M. Möser, R. Böhme, and D. Breuker, "An inquiry into money laundering tools in the bitcoin ecosystem," in *Proc. APWG eCrime Researchers Summit*, 2013, pp. 1–14.
[3] M. Bartoletti, B. Pes, and S. Serusi, "Data mining for detecting bitcoin Ponzi schemes," in *Proc. Crypto Valley Conf. Blockchain Technol. (CVCBT)*, 2018, pp. 75–84.
[4] N. Christin, "Traveling the silk road: A measurement analysis of a large anonymous online marketplace," in *Proc. 22nd Int. Conf. World Wide Web*, 2013, pp. 213–224.
[5] "Binance security breach update." Binance. 2019. [Online]. Available: https://www.binance.com/en/support/announcement/360028031711
[6] "Binance hack 2019—A deep dive into money laundering and mixing." Clain. 2019. [Online]. Available: https://clain.io/blog/post/binance-hack-2019-deep-dive-into-the-money-laundering

[7] J. Wu, J. Liu, W. Chen, H. Huang, Z. Zheng, and Y. Zhang, "Detecting mixing services via mining bitcoin transaction network with hybrid motifs," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 4, pp. 2237–2249, Apr. 2022.
[8] L. Wu et al., "Towards understanding and demystifying bitcoin mixing services," in *Proc. Web Conf.*, 2021, pp. 33–44.
[9] S. Farrugia, J. Ellul, and G. Azzopardi, "Detection of illicit accounts over the Ethereum blockchain," *Expert Syst. Appl.*, vol. 150, Jul. 2020, Art. no. 113318.
[10] M. Weber et al., "Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics," 2019, *arXiv:1908.02591*.
[11] D. Vassallo, V. Vella, and J. Ellul, "Application of gradient boosting algorithms for anti-money laundering in cryptocurrencies," *SN Comput. Sci.*, vol. 2, no. 3, pp. 1–15, 2021.
[12] H. Baek, J. Oh, C. Y. Kim, and K. Lee, "A model for detecting cryptocurrency transactions with discernible purpose," in *Proc. 11th Int. Conf. Ubiquitous Future Netw. (ICUFN)*, 2019, pp. 713–717.
[13] P. Nerurkar, S. Bhirud, D. Patel, R. Ludinard, Y. Busnel, and S. Kumari, "Supervised learning model for identifying illegal activities in bitcoin," *Appl. Intell.*, vol. 51, no. 6, pp. 3824–3843, 2021.
[14] O. Sagi and L. Rokach, "Ensemble learning: A survey," *Wiley Interdiscip. Rev. Data Min. Knowl. Disc.*, vol. 8, no. 4, 2018, Art. no. e1249.
[15] S. Ranshous et al., "Exchange pattern mining in the bitcoin transaction directed hypergraph," in *Proc. Int. Conf. Financ. Cryptogr. Data Security*, 2017, pp. 248–263.
[16] K. Toyoda, T. Ohtsuki, and P. T. Mathiopoulos, "Identification of high yielding investment programs in bitcoin via transactions pattern analysis," in *Proc. IEEE Global Commun. Conf.*, 2017, pp. 1–6.
[17] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Stat.*, vol. 29, no. 5, pp. 1189–1232, 2001.
[18] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, 2016, pp. 785–794.
[19] G. Ke et al., "LightGBM: A highly efficient gradient boosting decision tree," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–9.
[20] W. Zhaohui, S. Huawei, C. Wei, and C. Xueqi, "Survey on graph classification," *J. Softw.*, vol. 33, no. 1, pp. 171–192, 2021.
[21] N. M. Kriege, F. D. Johansson, and C. Morris, "A survey on graph kernels," *Appl. Netw. Sci.*, vol. 5, no. 1, pp. 1–42, 2020.
[22] "A guide to model stacking in practice." 2016. [Online]. Available: http://blog.kaggle.com/2016/12/27/a-kagglers-guide-to-model-stacking-in-practice
[23] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
[24] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, no. 85, pp. 2825–2830, 2011.
[25] G. Siglidis, G. Nikolentzos, S. Limnios, C. Giatsidis, K. Skianis, and M. Vazirgiannis, "GraKeL: A graph kernel library in Python," *J. Mach. Learn. Res.*, vol. 21, no. 54, pp. 1–5, 2020.
[26] F. Reid and M. Harrigan, "An analysis of anonymity in the bitcoin system," in *Security and Privacy in Social Networks*. New York, NY, USA: Springer, 2013, pp. 197–223.
[27] S. Meiklejohn et al., "A fistful of bitcoins: Characterizing payments among men with no names," in *Proc. Conf. Internet Meas. Conf.*, 2013, pp. 127–140.
[28] T. Liu et al., "A new bitcoin address association method using a two-level learner model," in *Proc. Int. Conf. Algorithms Archit. Parallel Process.*, 2019, pp. 349–364.
[29] J. Pakki, Y. Shoshitaishvili, R. Wang, T. Bao, and A. Doupé, "Everything you ever wanted to know about bitcoin mixers (but were afraid to ask)," in *Proc. Int. Conf. Financ. Cryptogr. Data Security*, 2021, pp. 117–146.
[30] I. Alarab, S. Prakoonwit, and M. I. Nacer, "Competence of graph convolutional networks for anti-money laundering in bitcoin blockchain," in *Proc. 5th Int. Conf. Mach. Learn. Technol.*, 2020, pp. 23–27.
[31] Y. Hu, S. Seneviratne, K. Thilakarathna, K. Fukuda, and A. Seneviratne, "Characterizing and detecting money laundering activities on the bitcoin network," 2019, *arXiv:1912.12060*.

Chang Xu received the Ph.D. degree in computer science from Beihang University, Beijing, China, in 2013.

She is currently an Associate Professor with the School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing. Her research interests include security and privacy in VANET, and big data security.

Ruting Xiong received the bachelor's degree from the School of Computer Science, China University of Geosciences, Wuhan, China, in 2021. She is currently pursuing the master's degree with the School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing, China.

Her research interests include blockchain security supervision and Bitcoin anomaly transaction detection.

Xiaodong Shen received the bachelor's and master's degrees from the School of Computer Science and Technology from Beijing Institute of Technology, Beijing, China, in 2015 and 2018, respectively, where he is currently pursuing the Ph.D. degree with the School of Cyberspace Science and Technology.

His current research interests include crowdsensing, security and privacy in IoT, and blockchain applications.

Liehuang Zhu (Senior Member, IEEE) received the Ph.D. degree in computer science from Beijing Institute of Technology, Beijing, China, in 2004.

He is currently a Professor with the School of Cyberspace Science and Technology, Beijing Institute of Technology. His research interests include security protocol analysis and design, group key exchange protocols, wireless sensor networks, and cloud computing.

Xiaoming Zhang received the B.Sc. degree and the M.Sc. degree in computer science and technology from the National University of Defense Technology, Changsha, China, in 2003 and 2007, respectively, and the Ph.D. degree in computer science from Beihang University, Beijing, China, in 2012.

He is currently with the School of Cyber Science and Technology, Beihang University, where he has been an Assistant Professor since 2012. He has published over 40 papers, such as *ACM Transactions on Information Systems*, IEEE TRANSACTIONS ON MULTIMEDIA, IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON CYBERNETICS, WWWJ, *Neurocomputing*, *Signal Processing*, ACM MM, AAAI, IJCAI, CIKM, ICMR, SDM, and EMNLP. His current research interests include social media analysis, image tagging, and text mining.