

# Detecting Mixing Services via Mining Bitcoin Transaction Network With Hybrid Motifs

Jiajing Wu<sup>1</sup>, Senior Member, IEEE, Jieli Liu<sup>1</sup>, Weili Chen, Huawei Huang<sup>1</sup>, Member, IEEE, Zibin Zheng<sup>1</sup>, Senior Member, IEEE, and Yan Zhang<sup>1</sup>, Fellow, IEEE

**Abstract**—As the first decentralized peer-to-peer (P2P) cryptocurrency system allowing people to trade with pseudonymous addresses, Bitcoin has become increasingly popular in recent years. However, the P2P and pseudonymous nature of Bitcoin make transactions on this platform very difficult to track, thus triggering the emergence of various illegal activities in the Bitcoin ecosystem. Particularly, *mixing services* in Bitcoin, originally designed to enhance transaction anonymity, have been widely employed for money laundering to complicate the process of trailing illicit fund. In this article, we focus on the detection of the addresses belonging to mixing services, which is an important task for anti-money laundering in Bitcoin. Specifically, we provide a feature-based network analysis framework to identify statistical properties of mixing services from three levels, namely, network level, account level, and transaction level. To better characterize the transaction patterns of different types of addresses, we propose the concept of attributed temporal heterogeneous motifs (ATH motifs). Moreover, to deal with the issue of imperfect labeling, we tackle the mixing detection task as a positive and unlabeled learning (PU learning) problem and build a detection model by leveraging the considered features. Experiments on real Bitcoin datasets demonstrate the effectiveness of our detection model and the importance of hybrid motifs including ATH motifs in mixing detection.

**Index Terms**—Anti-money laundering (AML), bitcoin, mixing services, network mining, network motifs.

## I. INTRODUCTION

**B**ITCOIN, the world’s first peer-to-peer (P2P) cryptocurrency system [1], has become one of the hottest buzzwords with a dominant share of the cryptocurrency market [2] due to its pseudonymous nature in decentralized trading process as well as its low transaction fees.

However, the P2P and pseudonymous nature of Bitcoin make transactions on this platform very difficult to track,

Manuscript received January 26, 2020; revised September 22, 2020; accepted December 29, 2020. This work was supported in part by the Key-Area Research and Development Program of Guangdong Province under Grant 2018B010109001; and in part by the National Natural Science Foundation of China under Grant 62032025, Grant 61973325, and Grant U1811462. This article was recommended by Associate Editor L. Sheremetov. (Corresponding author: Zibin Zheng.)

Jiajing Wu, Jieli Liu, Weili Chen, Huawei Huang, and Zibin Zheng are with the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510006, China (e-mail: wujiajing@mail.sysu.edu.cn; zhzibin@mail.sysu.edu.cn).

Yan Zhang is with the Department of Informatics, University of Oslo, 0316 Oslo, Norway.

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TSMC.2021.3049278>.

Digital Object Identifier 10.1109/TSMC.2021.3049278

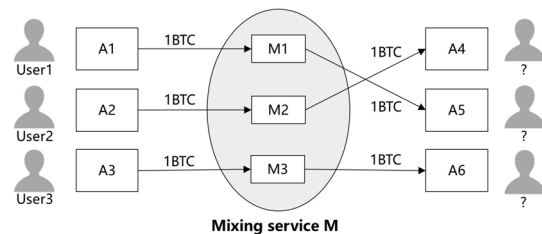


Fig. 1. Example of mixing services, which can conceal the identity of users and complicate fund tracing by participating in a transaction with multiple users.

thus triggering the emergence of various illegal activities in the Bitcoin ecosystem [3]. For instance, about 7000 Bitcoins which worth \$40.7 million were stolen from Binance recently [4], one of the largest cryptocurrency exchanges in the world. Then, the stolen Bitcoins can be cashed out directly through exchanges. However, before conducting the business, exchanges typically implement the know-your-customer (KYC) process, which is widely adopted in traditional e-payment scenarios to verify the identities of the users, review their financial activities, and ascertain what risks they may pose. With the enforcement of the KYC process, the identity of the thieves can be easily exposed via the identity information provided by the exchanges, and the stolen Bitcoins usually need to be laundered into “clean” Bitcoins by some techniques before being cashed out. It has been demonstrated that mixing services, such as BitLaundry, Helix Light, Bitcoin Fog, etc., have involved in this process of *money laundering* [5] and can be regarded as significant tools for concealing illicit profits in Bitcoin.

Bitcoin mixing services are originally designed to enhance the anonymity of transactions and make the sources of funds more untraceable. Fig. 1 gives a simple illustration of Bitcoin mixing. Three users represented as A1, A2, and A3 send one Bitcoin (abbreviation BTC) to three addresses M1, M2, and M3 of a mixing service M, respectively, and provide their own new addresses A4, A5, and A6 to receive the Bitcoin back. Then, M randomly selects an address from M1, M2, and M3 to return money to A4, A5, and A6. In this way, the relationships between sources and destinations are confused, thus increasing the difficulty of tracing the source of funds and analyzing the transaction behavior of users. Since Bitcoin is designed with pseudonymous identities and the real identity behind an address can be learned only when the user uses this address to buy or sell Bitcoins with an exchange, it is

unlikely to enforce the KYC process for regulation. Therefore, the study on the identification of mixing services and tracing illegal transactions in Bitcoin is of great value for building a healthier Bitcoin ecosystem.

Fortunately, the public and irreversible transaction records provide us an opportunity to detect irregular transaction patterns in Bitcoin. To this end, in this article, we focus on detecting addresses belonging to mixing services via mining the transaction records and attempt to characterize their transaction patterns. Based on the detection results, we can further chase up users involved in criminal activities by analyzing users who take part in Bitcoin mixing.

In recent years, several studies have shed light on the problem of detecting Bitcoin mixing services. It has been reported that mixing services and exchanges are two key components in laundering Bitcoins [5], [6] while mixing services have a higher propensity to be used in laundering illicit money [7]. To answer how mixing services work, the operation model of several mixing services was studied by reverse-engineering methods in [5]. Based on the observations given by [5], Prado-Romero *et al.* [8] proposed the problem of mixing detection and tackled this problem by exploiting the method of community outlier detection. Yet till now, Bitcoin mixing detection is still an extremely tricky task due to several great challenges as follows.

- 1) *Incomplete Label Information*: Labeled addresses associated with mixing services occupy only a small fraction of all addresses, and the true identities of most other addresses are unknown in Bitcoin.
- 2) *Dynamic Process With Multiple Transactions*: Some mixing services use hubs to combine multiple transactions or split a large amount of money into multiple smaller transactions, thus making it more difficult to identify the mixing processes as well as the addresses involved in the processes.
- 3) *Various Obfuscation Patterns*: Mixing services are provided by different third-party platforms, and their obfuscation patterns vary a lot from each other.

In this work, to deal with the problem of incomplete label information, we tackle the task of Bitcoin mixing detection as a positive and unlabeled learning (PU learning) problem [9] and then adopt a two-stage strategy to enhance the detecting performance. In order to analyze the transaction records more comprehensively, we construct two kinds of temporal directed transaction networks, including a homogeneous address–address interaction network (AAIN) and a heterogeneous transaction–address interaction network (TAIN), to depict the relationship between addresses and the relationship between addresses and transactions, respectively. Network motifs have been widely proven to be a powerful tool in handling various network mining tasks [10]–[12]. To better analyze the complicated dynamic processes in the Bitcoin transaction network, we propose a novel concept called attributed temporal heterogeneous motifs (ATH motifs) for the TAIN. The hybrid motifs composed of temporal homogeneous motifs in AAIN and ATH motifs in TAIN, are employed as the vital features for the detection of mixing services.

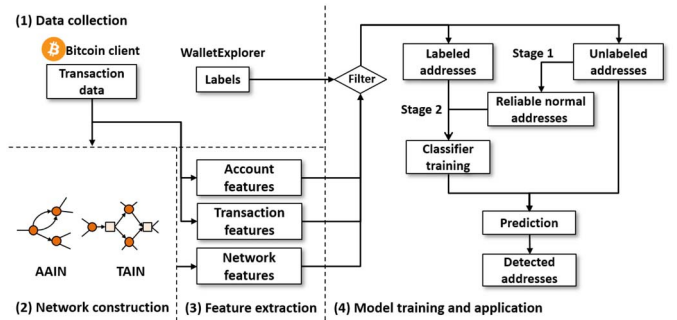


Fig. 2. Overview of the proposed Bitcoin mixing detection framework, including four modules, namely, data collection, network construction, feature extraction, and model training and application.

As shown in Fig. 2, the proposed mixing detection framework mainly contains four modules: 1) *data collection*, which gathers the Bitcoin transaction data from a Bitcoin client and crawls the label information from WalletExplorer;<sup>1</sup> 2) *network construction*, constructing AAIN and TAIN from the transaction records for feature extraction; 3) *feature extraction*, whose purpose is to extract features from multiple levels; and 4) *model training and application*, which trains the model using the training set, makes prediction for the unlabeled addresses and finally outputs the detected mixing addresses.

In summary, the main contributions of this article can be listed as follows.

- 1) To the best of our knowledge, we are the first to apply network motifs on the problem of Bitcoin mixing detection. We propose the novel concept of ATH motifs and demonstrate that both temporal and ATH motifs play an important role in Bitcoin mixing detection.
- 2) We propose a feature-based transaction network analysis framework and generalize the issue of Bitcoin mixing detection as a PU learning problem, the purpose being to make better use of the labeled addresses under the precondition of imperfectly labeled datasets.
- 3) The proposed model achieves a high true positive rate (TPR) and a low false positive rate (FPR) in Bitcoin mixing detection, which facilitates fund tracing and crime detection in the Bitcoin ecosystem.

The remaining sections of this article are organized as follows. Sections II–V introduce the details of the aforementioned four modules of the proposed mixing detection framework one by one. Then, we present experimental results in Section VI. Finally, we provide some related work in Section VII and conclude this article in Section VIII.

## II. DATA COLLECTION

WalletExplorer is a smart Bitcoin block explorer providing label information of addresses by making transactions with some services and observing how the Bitcoin flows merge. However, the name database of WalletExplorer no longer updated since 2016, which means that WalletExplorer does not include the new emerging services. The transaction data

<sup>1</sup><https://www.walletexplorer.com>

TABLE I  
STATISTICS OF THE DATASETS

Dataset	Start time	Unlabeled address	Labeled address <sup>1</sup>		
			BitcoinFog	BitLauder	HelixMixer
2014	00:00, Nov. 1	2,507,872	6088	8	0
2015	00:00, Jun. 1	2,525,038	3911	9	2
2016	00:00, Jan. 1	2,502,738	198	2	3856

<sup>1</sup> Addresses of three mixing services including Bitcoin Fog, BitLauder and Helix Mixer crawled from WalletExplorer are as our labeled addresses.

of Bitcoin are contained in blocks orderly, and they are publicly accessible by running a Bitcoin client. Considering the sufficiency of labeled addresses for training and the huge volume of Bitcoin transaction records, we extract three snapshots of Bitcoin transaction data between November 2014 and January 2016 with six months being the sampling interval. Each snapshot contains 1 500 000 transaction records. And we crawl the labeled addresses belonging to mixing services from WalletExplorer. The three snapshots with label information are referred to as the 2014, 2015, and 2016 datasets.

The labeled addresses obtained from WalletExplorer are mainly belonging to three mixing services as follows.

- 1) Bitcoin Fog<sup>2</sup> is one of the earliest mixing services and can be accessed only via Tor. When using this service, each withdrawal will be split into a random number of transactions spreading out randomly over a specific time period.
- 2) BitLauder announced that it was “the best Bitcoin laundry and Bitcoin laundering service.” However, it has been reported as one of the weakest mixers of all tested in the analysis conducted by de Balthasar and Hernandez-Castro [13]. Unfortunately, the detailed information of this service is not available anymore.
- 3) Helix (has been offline since 2017)<sup>3</sup> offered two versions of mixing services, including a standard version and a light version. The standard version required their users to register a wallet, and then the Bitcoins sent to the wallet would be automatically mixed and finally sent to a defined address. While for the light version, the Bitcoins could be withdrawn to up to five addresses.

Table I shows the statistics of these three datasets. On the average, the labeled addresses only account for about 0.19% of all addresses appearing in the transaction data.

### III. NETWORK CONSTRUCTION AND MOTIF DEFINITION

Transaction records of Bitcoin can be abstracted as a huge complex network, where each node refers to a Bitcoin address and each edge represents a transaction process between addresses. Using this simple modeling method, we construct a homogeneous AAIN to investigate the interaction patterns of addresses. Since Bitcoin transactions usually involve multiple inputs and multiple outputs, from this figure depicting the transaction relationships between address pairs, it is difficult to figure out how much an address has taken from another address. To this end, we construct a heterogeneous TAIN

<sup>2</sup><https://bitcointalk.org/index.php?topic=50037.0>

<sup>3</sup><https://bitcointalk.org/index.php?topic=5238537.0>

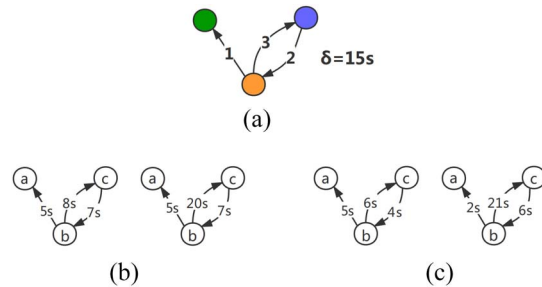


Fig. 3. Example of a (a) motif and (b) its instances. Graph patterns in (c) are not instances of  $M$  because of their out of order edge sequence or their out of range edge occurring time (the constrained time window  $\delta$  is set as 15 s).

to represent the transaction amount information. This is an attributed temporal heterogeneous information network (HIN) where a node can be either a particular transaction or an address. From TAIN, we can clearly find how much an address has been sent to or received from a transaction. Therefore, compared with AAIN, TAIN can display the strength of money transfer more clearly. Network motifs, which can be regarded as the recurrent small subgraph patterns in networks, have been demonstrated as an important tool for characterizing higher-order interactions and understanding various properties of complex systems [10], [11]. In the following, we present the definition of AAIN, TAIN, and their motifs in detail.

#### A. AAIN and Temporal Motifs

*Definition 1 (AAIN):* An AAIN is a temporal direct multi-graph  $G = (V, E)$ , where  $V$  is the set of nodes and  $E$  is the set of edges carrying temporal information. Each node  $v \in V$  denotes a Bitcoin address and each edge  $e \in E$  standing for a transaction is defined as a tuple  $(u, v, tx, t)$ , denoting that address  $u$  is a source and address  $v$  is a destination for a transaction  $tx$  happening at time  $t$ .

*Definition 2 (Temporal Motifs):* Temporal motifs are defined as recurring interconnection patterns occurring in temporal networks [11]. Particularly, a  $k$ -node,  $l$ -edge,  $\delta$ -temporal motif instance  $M_{\delta}^{k,l}(G)$  of a temporal network  $G = (V, E)$  can be represented as

$$M_{\delta}^{k,l}(G) = (V_M^k, E_M^l, \delta)$$

where  $V_M^k$  ( $V_M^k \subseteq V$ ) is a set of  $k$  nodes,  $E_M^l$  ( $E_M^l \subseteq E$ ) is a set of  $l$  edges, and  $\delta$  is a time window indicating that all of edges in the motif occur within a  $\delta$  duration, i.e., an increased sequence  $t_1, t_2, \dots, t_l$  which records the timestamp of each edge in the motif instance satisfies  $t_1 \leq t_2 \leq \dots \leq t_l$  and  $t_l - t_1 \leq \delta$ .

Different from static network motifs, temporal motifs well preserve the time-ordered sequence of contacts in a time window, being effective in analyzing the temporal structure of complex networks. Fig. 3(a) and (b) illustrates a 3-node, 3-edge,  $\delta$ -temporal motif  $M$  and its instances, while graph patterns in Fig. 3(c) are not instances of  $M$  because their edge order or occurring time window does not satisfy the condition.

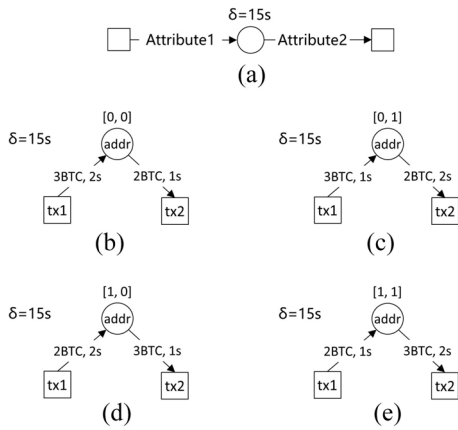


Fig. 4. (b)–(e) Four instances of the (a) ATH motif in the transaction network. The edge vector  $\Gamma_V$ , which maps the amount value attribute to the first bit and the time information to the second bit, can differentiate varieties of transaction patterns with the same topology. In this case, the first bit of  $\Gamma_V$  is set as 0 if the amount of  $tx_1$  is higher than  $tx_2$ , and as 1 otherwise; and the second bit of  $\Gamma_V$  is set as 0 if the time of  $tx_1$  is later than  $tx_2$ , and as 1 otherwise.

### B. TAIN and ATH Motifs

**Definition 3 (TAIN):** A TAIN is an attributed temporal HIN  $G = (V, E, \Omega)$  with  $\varphi_V: V \rightarrow \{\text{address}, \text{transaction}\}$  for node-type mapping,  $\varphi_E: E \rightarrow \{\text{transaction-address}, \text{address-transaction}\}$  for edge-type mapping, and  $\Omega$  denoting the set of attributes attached to edges in the graph, including transaction amount and transaction time. A *transaction–address* edge  $(u, tx_{in}, a, t_1)$  denotes that an input transaction  $tx_{in}$  happens at time  $t_1$  and transfers  $a$  Bitcoins into an address  $u$ , while an *address–transaction* edge  $(v, tx_{out}, b, t_2)$  denotes that an output transaction  $tx_{out}$  happens at time  $t_2$  and transfers  $b$  Bitcoins out of an address  $v$ .

**Definition 4 Attributed Temporal Heterogeneous (ATH) Motifs:** ATH motifs are local recurring subgraphs of attributed temporal HINs, described by a set of nodes, a set of edges, attributes, and a time window. A  $\delta$ -ATH motif instance of an attributed temporal HIN  $G = (V, E, \Omega)$  can be defined as

$$M_{ATH}^\delta(G) = (V_{ATH}, E_{ATH}, \Gamma_\Omega, \delta)$$

where  $V_{ATH}$  ( $V_{ATH} \subseteq V$ ) represents the set of nodes, and  $E_{ATH}$  ( $E_{ATH} \subseteq E$ ) represents the set of edges, also satisfying node types  $|\{\varphi_V(v) | v \in V_{ATH}\}| > 1$  or edge types  $|\{\varphi_E(e) | e \in E_{ATH}\}| > 1$ .  $\Gamma_\Omega$  denotes a mapped vector of edge attributes, and  $\delta$  is a time window that constrains  $\min(\psi_E(e)) + \delta \leq \max(\psi_E(e))$  for  $e \in E_{ATH}$ , where  $\psi_E$  is a time mapping which maps each edge  $e \in E$  to its occurring time.

Though some subgraphs in the TAIN may share the same topology, the attribute information can make them different. Taking the ATH motif shown in Fig. 4 and its four instances as an example, the instance in Fig. 4(c) represents the transaction pattern of receiving money first and sending out less money later, while the instance in Fig. 4(d) stands for sending money out first and receiving less money later, which has an opposite transaction order and leads to a negative balance.

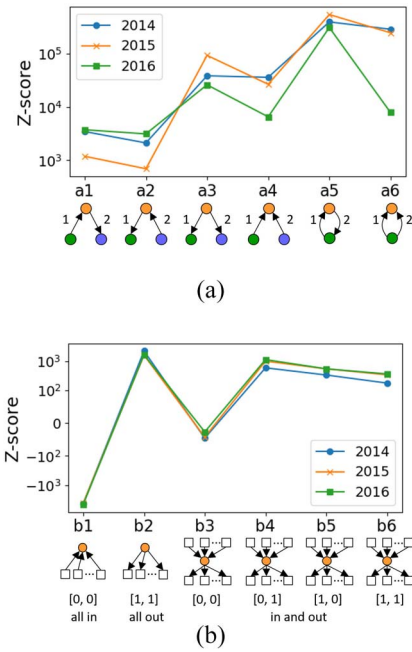


Fig. 5. Z-score value of the candidate motif patterns. (a) Candidate 2-edge  $\delta$ -temporal motif patterns in AAIN. (b) Candidate  $\delta$ -ATH motif patterns in TAIN.

## IV. FEATURE-BASED ANALYSIS

Due to the specific function of mixing Bitcoins, addresses associated with mixing services may have several unique features different from normal addresses. In the following, we aim to extract features of the addresses from three levels and conduct descriptive statistics on them.

### A. Network Features

Different types of objects have different interaction ways in complex systems, which would affect the topological structure of the whole network. In this part, we extract network features from both AAIN and TAIN. To characterize the interaction patterns and reveal the functional properties in the network, we propose to take some higher-order network features (i.e., network motifs) into account.







Network motifs are small subgraph patterns in a network that occurring with significantly higher frequency than those in randomized networks [10]. The statistical significance of a pattern can be measured by  $z$ -score, calculated as

$$z\text{-score} = \frac{n_{\text{real}} - \bar{n}_{\text{random}}}{\text{std}(n_{\text{random}})} \quad (1)$$

where  $n_{\text{real}}$  denotes the frequency of the pattern occurring in the real network, and  $\bar{n}_{\text{random}}$  and  $\text{std}(n_{\text{random}})$  denote the mean and variance of the pattern occurring frequencies in a set of random networks. A pattern is usually regarded as a statistically significant motif if its  $z$ -score  $> 2.0$  [14].

We consider six simplest transaction patterns [Fig. 5(a)] with two edges in AAIN, which illustrate how an address interacts with other addresses for a  $\delta$  duration. For example, pattern  $a1$  represents that an address first receives money from a neighbor and then transfers money to another neighbor while

TABLE II  
AVERAGE FRACTION OF  $\delta$ -TEMPORAL MOTIFS ( $\delta = 3$  H)

Temporal motif						
Labeled address	0.2552	0.0051	0.5902	0.1465	0.0000	0.0030
Unlabeled address	0.2320	0.0576	0.4016	0.2395	0.0003	0.0690

pattern  $a2$  represents an opposite transaction order. Moreover, we abstract the transactions of an address occurring within a  $\delta$  time window as three kinds of topological structures in TAIN: *all in*, *all out*, and *in and out*, which illustrate only input transactions, only output transactions, and both input and output transactions occurring within the time window, respectively. By taking the amount information and temporal information into account, the direction and strength of Bitcoin transfer can be better reflected in these substructures. In our scenario, the relative size between the attribute information of input and output transactions is more important than their actual absolute value. Therefore, the *in* and *out* structure can be further divided into four patterns, and all these six patterns [Fig. 5(b)] can be represented as candidate  $\delta$ -ATH motifs with a binary value function defining each bit of  $\Gamma_{\Omega}$  as

$$\Gamma_{\Omega}[0] = \begin{cases} 0 & \bar{v}_{in} \geq \bar{v}_{out} \\ 1 & \bar{v}_{in} < \bar{v}_{out} \end{cases}, \quad \Gamma_{\Omega}[1] = \begin{cases} 0 & \bar{t}_{in} > \bar{t}_{out} \\ 1 & \bar{t}_{in} \leq \bar{t}_{out} \end{cases} \quad (2)$$

where  $\Gamma_{\Omega}[0]$  and  $\Gamma_{\Omega}[1]$  denote the first and the second bit of the mapped vector, respectively,  $\bar{v}_{in}$  and  $\bar{v}_{out}$  denote the average amount value of input transactions and output transactions, respectively, and  $\bar{t}_{in}$  and  $\bar{t}_{out}$  denote the average time of input transactions and output transactions, respectively. Particularly, for the *all in* patterns which has no value of  $\bar{v}_{out}$  and  $\bar{t}_{out}$ , the corresponding  $\Gamma_{\Omega}$  is defined as  $[0, 0]$ . Contrarily,  $\Gamma_{\Omega}$  of *all out* patterns are given as  $[1, 1]$ .

The  $z$ -score value of these patterns (Fig. 5) are calculated with 100 random networks for each real network. To preserve the same degree sequence and attribute distribution as the real network, these random networks are generalized by the configuration model [15] with a rearrangement of the attribute information. As displayed in Fig. 5, the occurring frequency of a pattern is similar in different networks since these networks are from the same domain. Patterns  $a1$ – $a6$  and patterns  $b2$  and  $b4$ – $b6$  are statistically significant motifs in the Bitcoin transaction network with a much greater number of occurring times in a random network. We then make use of these hybrid motifs, including temporal motifs  $a1$ – $a6$  and ATH motifs  $b2$  and  $b4$ – $b6$  to characterize the network features in the Bitcoin transaction network. Besides, we extract some basic network features from AAIN, such as in-degree, out-degree, and so on. All the network features are described as follows.





**NF1–NF6:** The occurring frequency proportion of each temporal motif in all considered temporal motifs in the first-order AAIN for each address.

**NF7–NF10:** The frequency proportion of an address being a part of each ATH motif in all considered ATH motifs.

**NF11:** The value of the in-degree in AAIN.

**NF12:** The value of the out-degree in AAIN.

TABLE III  
AVERAGE FRACTION OF  $\delta$ -ATH MOTIFS ( $\delta = 3$  H)

ATH motif				
Labeled address	0.4957	0.4916	0.0057	0.0069
Unlabeled address	0.6557	0.3148	0.0069	0.0225

**NF13:** The ratio of the in-degree to the out-degree in AAIN.  
**NF14:** The number of unique successor addresses in AAIN.  
**NF15:** The number of unique predecessor addresses in AAIN.

**NF16:** The ratio of the in-degree to the number of unique successor addresses in AAIN.

**NF17:** The ratio of the out-degree to the number of unique predecessor addresses in AAIN.

Tables II and III describe the average value for NF1–NF10 in labeled addresses and unlabeled addresses with  $\delta = 3$  h, respectively. The selected time window will be explained in Section IV-C. And by combining with the statistics of NF11–NF17 in Table IV, we can summarize several findings from network features as follows.

**Finding 1:** The average fraction of  $a1$  pattern is much higher than the fraction of  $a2$  pattern, and the fraction of  $b4$  pattern far outstrips the other kinds of *in and out* motifs  $b5$  and  $b6$ . Besides, this kind of difference is more significant for labeled addresses. In other words, mixing services are more in line with the transaction pattern of receiving money first and sending money out later with a balance not less than 0.

**Finding 2:** Based on the results of  $a5$  and  $a6$  patterns, we can conclude that nonmixing service entity may reuse some addresses in a short time while this situation seldom happens for mixing services. Besides, the statistics of NF16 and NF17 also indicates the same finding.

**Finding 3:** The prevalence of  $a3$  pattern is due to the change addresses generated to receive the change. Besides, from the fraction of  $a3$  as well as the statistics of NF11–NF13, we can see that mixing services prefer dispersing the tainted Bitcoins to others, which is a usually adopted strategy for Bitcoin mixing. Some analytic companies apply some taint analysis techniques [16], which can predict a risk score for addresses and blacklist the high-risk tainted coin possessors, to track these patterns and avoid buying these tainted coins.

## B. Account Features

The state and activeness of an address, in many cases, may reflect which category the address belongs to and, thus, we introduce account features to describe the state and activeness of an address. For example, addresses belonging to Bitcoin

TABLE IV  
STATISTICS OF FEATURES (EXCEPT MOTIF FEATURES)

	NF11	NF12	NF13	NF14	NF15	NF16	NF17	AF1	AF2	AF3	AF4	AF5	AF6	TF1	TF2	TF3	TF4	TF5	TF6
Labeled address																			
Mean	5.90	9.02	0.85	5.86	8.71	1.01	1.02	1.28	1.28	1.00	1.48	1.48	1.00	0.00	5.59e+3	10.20	5.34	11.94	6.22
StdDev	25.31	23.52	2.52	25.11	15.27	0.18	0.13	2.56	2.54	0.03	7.22	7.21	0.45	0.27	2.55e+4	7.83	25.46	13.89	32.80
Median	2.00	10.00	0.48	2.00	10.00	1.00	1.00	1.00	1.00	1.00	0.32	0.32	1.00	0.00	1.36e+3	7.00	2.00	8.00	2.00
Unlabeled address																			
Mean	6.86	8.28	1.30	6.23	6.63	1.12	1.29	1.79	1.87	1.02	6.79	6.80	57.50	0.10	3.80e+4	31.65	59.65	38.68	132.39
StdDev	91.47	222.59	5.21	43.54	75.23	5.15	5.45	42.93	43.07	0.56	1.23e+3	1.23e+3	6.88e+4	15.63	1.10e+5	95.78	375.36	129.82	833.56
Median	2.00	2.00	0.76	2.00	2.00	1.00	1.00	1.00	1.00	1.00	0.11	0.11	1.00	0.00	3.31e+4	3.00	2.00	3.00	2.00

exchanges usually have a higher trade frequency for a great many of businesses, while the trade frequency of many ordinary users is relatively much lower. The extracted account features for each address, referred to as AFs, are detailed as follows.

*AF1*: The number of input transactions in the snapshot.

*AF2*: The number of output transactions in the snapshot.

*AF3*: The number ratio of the input transactions to the output transactions.

*AF4*: The total amount of input transactions in the snapshot.

*AF5*: The total amount of output transactions in the snapshot.

*AF6*: The amount ratio of the total input transactions to the total output transactions.

From the statistics of AF1–AF6 in Table IV, some notable results can be obtained as follows.

*Finding 4*: There exists a large variety among the unlabeled addresses in terms of account features because there exist multiple types of unlabeled addresses. However, the difference of account features between labeled addresses is relatively smaller, as illustrated by a relatively low value of standard deviation.

*Finding 5*: According to the results in terms of AF6, the amount value of output transactions usually equals to that of input transactions for labeled addresses, while the unlabeled addresses keep a positive net income on average, which can fully illustrate that addresses belonging to mixing services act like intermediaries by sending out what they have received.

### C. Transaction Features

Next, the transaction behaviors of addresses in the mixing process are measured by transaction features. Since the relationship between senders and recipients of a mixing process would be obviously detected if the mixing service directly sends out an approximate equal amount to its recipients in the following blocks, mixing services may use many addresses acting like “intermediary” addresses (e.g., hubs) to participate in the process of fund splitting and integrating [5]. After spread by a large number of intermediary addresses over a period of time, Bitcoins from the transaction sources are finally sent to the corresponding recipients.

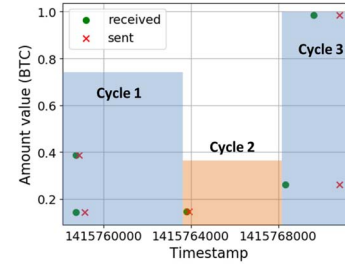


Fig. 6. Transaction cycle is composed of a continuous input stream and a continuous output stream. Three transaction cycles of labeled address “1NsNkSxyYjB9o3QkPT2RjTXST4nGRtfMzS” are shown in this figure.

Here, we introduce the concept of *transaction cycle*, consisting of an ordered pair of continuous input and output streams, to describe the process of money flowing through an intermediary address for Bitcoins enrolled in mixing services. Fig. 6 displays ten continuous transactions of an address belonging to Bitcoin Fog. The  $x$ -axis in this figure represents the time line, while the  $y$ -axis represents how much the address received and sent. These ten transactions are distributed in three transaction cycles, and during each cycle, the address finally sent out what it had received with an increased balance value 0. We observe that many labeled addresses have similar transaction behavior like this, and suppose that this behavior is associated with the nature of being an intermediary. Several transaction features (referred as TFs) extracted to describe the transaction behaviors of an address are as follows.

*TF1*: The standard deviation of the increased balance in every transaction cycle (the expected value of increased balance for an intermediary address is 0).

*TF2*: The average time interval  $\bar{T}$  between the first input transaction and the last output transaction in each cycle.

*TF3*: The average number of addresses that jointly participate as the inputs of a transaction.

*TF4*: The average number of addresses that jointly participate as the outputs of a transaction.

*TF5*: The total number of unique addresses that jointly participate as the inputs.

*TF6*: The total number of unique addresses that jointly participate as the outputs.

We obtain some observations based on the statistics of TF1–TF6 in Table IV.

*Finding 6*: Combining with the cumulative proportion line chart of  $\bar{T}$  shown in Fig. 7, we can observe

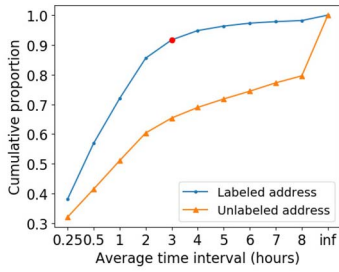


Fig. 7. Cumulative proportion of average time  $\bar{T}$ .

that the average time interval of transaction cycles of mixing services is mostly within 3 h, while the transaction cycle duration of an unlabeled address does not have such an obvious pattern. One possible explanation of this phenomenon is that mixing services are designed to process Bitcoins within a specific time as they are user-oriented services. As a result, we preliminarily set  $\delta = 3$  h as the time window of motifs when conducting descriptive statistics and mixing detection.

*Finding 7:* The TF1 results of addresses belonging to mixing services are closer to 0, indicating that the balance of these addresses over each transaction cycle is closer to 0. Besides, according to the statistics of TF3–TF6, these addresses have more co-input addresses than co-output addresses, which may be explained by that the coins of an address belonging to a mixing service are often obfuscated with coins of other addresses.

## V. DETECTION MODEL

In the mixing detection task considered here, we only access a small number of verified labeled addresses belonging to mixing services while the rest addresses are unlabeled. This problem of extreme class imbalance may greatly hinder the performance of supervised classification. To deal with this problem, we develop a positive and unlabeled (PU) learning model with a two-stage strategy. The first stage is to select out the reliable negative instances from the unlabeled instances (unlabeled addresses) in the training set, and the second stage is to train a classifier with the positive instances (labeled addresses) as well as the reliable negative instances in the training set.

*In stage one,* according to the spy technique proposed in [9], we sample a set of spy instances from the positive instances with a default sample rate 15%. The rest of the positive instances is set with label 1, while the spy instances as well as the unlabeled instances are set with label  $-1$ , and then they are used to train a classifier for selecting out the reliable negative instances. Here, we employ the widely considered logistic regression (LR) as the classifier. Since the spy instances are actually positive instances, the probability of a spy being predicted as a positive instance would be usually

higher than that of a negative instance. Therefore, we can select a threshold  $\theta$  based on the prediction probabilities of the spy instances, and the reliable negative instances are selected out from the unlabeled instances if their probability of being predicted as a positive instance is lower than  $\theta$ . The threshold  $\theta$  is selected as the value that can maximize the increment difference between the cumulative proportion of unlabeled instances and spy instances under a minute increment  $\Delta p$ , and it can be calculated by

$$\theta = \arg \max_{p \in [0 + \Delta p, 1]} (\Delta F_U(p) - \Delta F_S(p)). \quad (3)$$

For each instance  $i$  in the instance set, its probability of being predicted as a positive instance is denoted as  $x_i$  and stored in a set  $X$ , namely,  $x_i \in X$ . For the set  $X$ , its cumulative distribution function  $F(\cdot)$  is given by  $F_X(p) = P\{X \leq p\}$ , where  $P\{X \leq p\}$  represents the probability that a value in  $X$  is lower than or equal to a value  $p$ . Then, the increment of  $F_X(p)$  under a minute increment  $\Delta p$  ( $\Delta p = 0.005$  in our model) is denoted as  $\Delta F_X(p) = F_X(p) - F_X(p - \Delta p)$ . We use  $S$  and  $U$  to denote a set storing the prediction probabilities of spy instances and unlabeled instances, respectively, so that  $\Delta F_S(p)$  and  $\Delta F_U(p)$  represent the increment of  $F_S(p)$  and the increment of  $F_U(p)$  under the increment  $\Delta p$ , respectively.

*In stage two,* with the consideration that the number of positive instances and that of reliable negative instances may be imbalanced, we set different penalty weights for different kinds of instances in the loss function. The following objective function should be minimized:

$$C_+ \sum_{y_i=1} l(y_i, f(\mathbf{x}_i)) + C_- \sum_{y_i=-1} l(y_i, f(\mathbf{x}_i)) + \lambda R(\mathbf{w}) \quad (4)$$

where  $C_+$  and  $C_-$  denote the penalty coefficients of positive and reliable negative instances, respectively,  $l(y_i, f(\mathbf{x}_i))$  is the loss term,  $R(\mathbf{w})$  is the regularization term, and  $\lambda$  is the regularization coefficient. In this work, we apply a biased LR classifier so that the loss term is set to be log loss and the regularization term is set to be  $L2$ -norm. Besides,  $C_+$  and  $C_-$  are inversely proportional to the number of positive and reliable negative instances in our settings.

Finally, we choose a probability threshold  $\varepsilon$  and make a decision according to the prediction probability of each unlabeled address. An unlabeled address is detected as an address associated with mixing services when its probability of being predicted as a positive instance is greater than  $\varepsilon$ .

## VI. EXPERIMENTAL RESULTS

In this section, we conduct a comprehensive evaluation of the proposed detection framework for Bitcoin mixing services. First, we describe our experimental settings. Second, we present the experimental results of the proposed method in comparison with several baseline methods. After that, the effects of motif-based features and other basic features are compared and summarized. Next, we demonstrate the robustness of our framework via a parameter sensitivity analysis. Finally, since our experiments are conducted on three transaction snapshots during 2014–2016, we discuss about how these data are relevant for current transactions and addresses.

TABLE V  
PERFORMANCE COMPARISON OF DIFFERENT METHODS (WITH STANDARD DEVIATION)

Dataset	Metric	OCSVM	IF	LR	DT	IS1 <sup>1</sup>	IS2 <sup>2</sup>	Our method
2014	TPR	0.8986±0.0080	0.8991±0.0084	0.1681±0.0107	0.7052±0.0162	0.8265±0.0061	0.8265 ±0.0061	<b>0.9165±0.0060</b>
	FPR	0.2026±0.0064	0.1285±0.0139	0.0±0.0*	0.0±0.0*	0.0406±0.0003	0.0363±0.0003	<b>0.0334±0.0010</b>
	G-Mean	0.8465±0.0033	0.8851±0.0068	0.4098±0.0130	0.8397±0.0097	0.8905±0.0033	0.8924±0.0033	<b>0.9412±0.0029</b>
2015	TPR	0.8972±0.0105	0.8996±0.0106	0.0598±0.0096	0.6210±0.0245	0.7832±0.0112	0.7832±0.0112	<b>0.9149±0.0081</b>
	FPR	0.1900±0.0124	0.1598±0.0147	0.0±0.0*	0.0±0.0*	0.0438±0.0003	0.0448±0.0003	<b>0.0379±0.0016</b>
	G-Mean	0.8524±0.0047	0.8693±0.0075	0.2437±0.0197	0.7879±0.0156	0.8654±0.0062	0.8650±0.0062	<b>0.9382±0.0038</b>
2016	TPR	0.8953±0.0105	0.9005±0.0105	0.0004±0.0005	0.3916±0.0388	<b>0.9388±0.0061</b>	<b>0.9388±0.0061</b>	0.9318±0.0066
	FPR	0.1652±0.0106	0.2224±0.0273	0.0±0.0*	0.0±0.0*	0.0591±0.0004	0.0586±0.0003	<b>0.0356±0.0010</b>
	G-Mean	0.8645±0.0051	0.8366±0.0135	0.0115±0.0150	0.6250±0.0314	0.9398±0.0031	0.9400±0.0031	<b>0.9479±0.0031</b>

<sup>1,2</sup> IS1 and IS2 use two different inter links counting function, namely relative inter links and total inter links respectively [8].

\* The marked FPRs imply that there exist overfitting problems in LR and DT as the positive instances are more likely to be predicted as negative instances.

### A. Experimental Settings

We initialize the time window  $\delta = 3$  h and the probability threshold  $\varepsilon = 0.6$ . All the reported results are averaged over 100 independent experiments with the standardized features as the model inputs.

*Datasets:* As mentioned in Section II, we obtain three datasets with transaction data from a Bitcoin client as well as labels from WalletExplorer. Before training the model, we filter out the addresses with either only input transactions or output transactions. By applying this simple rule, 131 labeled addresses and 1 635 904 unlabeled addresses are filtered from the three datasets, occupying 0.9% and 22.2% of their corresponding class, respectively. This operation is based on the following considerations. On the one hand, mixing services serve as intermediaries in obfuscating the transactions so that addresses with either only input transactions or output transactions do not satisfy this obvious feature. On the other hand, 96.2% of the filtered labeled addresses have only one transaction record, which are not suitable for feature learning. Besides, the timestamp of all the transactions related to the filtered labeled addresses included in our datasets are close to the time boundaries of the snapshot datasets. Thus, there may be some extra transactions not being captured in our three snapshots. We then divide each dataset into the training set and the testing set as follows, and for each dataset, we train a model with the training set and verify the model with the testing set.

- 1) *Training Set:* For stage one, we select 70% unlabeled addresses and 70% labeled addresses to form the training set, and then we can obtain some reliable negative instances. For stage two, the training set is made up of 70% reliable negative instances as well as the labeled addresses used in stage one.
- 2) *Testing Set:* The testing set is formed by the remaining 30% reliable negative instances and 30% labeled addresses to evaluate our model.

*Evaluation Metrics:* In this work, we evaluate the performance of our model in terms of TPR, FPR, and the geometric mean (G-Mean). G-Mean was suggested in [17] and has been widely used as a comprehensive metric in evaluating classification performances on imbalanced datasets [18], [19]. Taking both the accuracy of positive instances and negative

instances into account, G-Mean is defined as follows:

$$\text{G-Mean} = \sqrt{\text{TPR} \times (1 - \text{FPR})}. \quad (5)$$

### B. Method Comparison

Our model is based on PU learning with a two-stage strategy, which is actually a semisupervised learning method. To evaluate the effectiveness of PU learning in our scenario, we compare our model with several baseline methods, including one-class support vector machine (OCSVM), isolation forest (IF) [20], LR, decision tree (DT), and InterScore (IS) [8]. Among them, OCSVM and IF are two unsupervised anomaly detection method, LR and DT are two widely used supervised classifiers. IS is a Bitcoin mixing detection method which can detect mixing service entities containing multiple addresses with community anomaly detection, as addresses belonging to these entities usually have more intercommunity connections than other addresses. Since here we focus on the problem of detecting addresses of mixing services, we consider the label of an address is equal to the label of its entity when implementing IS.

Table V compares the performance of our method with the baseline methods. Specifically, the proportion of outliers in the datasets is set to be 10% when fitting OCSVM and IF. According to Table V, we have the following observations.

- 1) The unsupervised anomaly detection methods (i.e., OCSVM, IF, and IS) can discover most of the positive instances, however, they have a higher FPR than other methods. In particular, since IS only captures one important topology feature of being an intermediary in user transactions, it is in lack of generalization so that its performance is significantly differentiated in different datasets.
- 2) The two supervised methods, including LR and DT, lead to the problem of overfitting and relatively poor performance. There exist two possible reasons for this result, one reason is that the extreme class imbalance hinders the performance of supervised classification, and the other reason is that these two methods treat all unlabeled addresses as negative instances, which may induce noises to the datasets.



TABLE VI  
PERFORMANCE COMPARISON OF DIFFERENT FEATURES (WITH STANDARD DEVIATION)

Dataset	Metric	Basic features	Temporal motifs	ATH motifs	Hybrid motifs*	Basic features & Temporal motifs	Basic features & ATH motifs	Basic features & Hybrid motifs*
2014	TPR	0.8744±0.0145	0.8728±0.0070	0.7059±0.0111	0.8912±0.0064	0.9032±0.0064	0.8797±0.0089	<b>0.9165±0.0060</b>
	FPR	0.1779±0.0128	0.0455±0.0009	0.1508±0.0013	<b>0.0318±0.0007</b>	0.0362±0.0014	0.1350±0.0091	0.0334±0.0010
	G-Mean	0.8479±0.0120	0.9127±0.0036	0.7742±0.0059	0.9289±0.0032	0.9330±0.0033	0.8723±0.0075	<b>0.9412±0.0029</b>
2015	TPR	0.8146±0.0115	0.8453±0.0098	0.8426±0.0100	0.8823±0.0088	0.8864±0.0092	0.8543±0.0095	<b>0.9149±0.0081</b>
	FPR	0.1388±0.0038	0.1423±0.0024	0.0716±0.0009	0.0667±0.0020	0.0878±0.0079	0.0852±0.0018	<b>0.0379±0.0016</b>
	G-Mean	0.8376±0.0064	0.8515±0.0043	0.8845±0.0051	0.9074±0.0041	0.8992±0.0065	0.8840±0.0048	<b>0.9382±0.0038</b>
2016	TPR	0.6442±0.0317	0.9271±0.0073	0.6639±0.0145	0.9123±0.0071	<b>0.9335±0.0067</b>	0.8150±0.0112	0.9318±0.0066
	FPR	0.3812±0.0077	0.0584±0.0012	0.3154±0.0043	0.0356±0.0011	0.0508±0.0011	0.1995±0.0047	<b>0.0356±0.0010</b>
	G-Mean	0.6311±0.0129	0.9343±0.0035	0.6741±0.0061	0.9380±0.0034	0.9413±0.0031	0.8077±0.0047	<b>0.9479±0.0031</b>

\* Hybrid motifs are a combination of Temporal and ATH motifs.

3) By selecting reliable negative instances from unlabeled instances first and then apply a supervised method, the proposed strategy can improve the detection rate of positive instances compared with directly applying supervised approaches, and obtain the best results in terms of G-Mean.

These observations show that the PU learning framework performs better on Bitcoin mixing detection with a high TPR exceeding 91% and a low FPR below 4% on extremely imbalanced datasets.

### C. Feature Performance Comparison

To verify the effectiveness of the proposed motif features, we divide all the features given in Section IV into basic features (features except motifs) and motifs (including temporal and ATH motifs) to train the classifier and further evaluate the importance of motifs in the detection. A detailed comparison is given in Table VI and can be summarized as follows.

- 1) The detection performance is relatively poor when we only use the basic features. While network motifs, which can reveal the higher-order features in a complex network, achieve decent performance in mixing detection.
- 2) Each evaluation metric can be significantly improved for almost all cases when combining hybrid motifs with basic features, which demonstrates that hybrid motifs play an indispensable role in the task of Bitcoin mixing detection.

Additionally, since LR has a good explainability in its model weights, which can reflect the influence degree of different features to the detection result, we analyze the impact of the features according to their absolute value of weight averaged over the three datasets in the 100 independent experiments. We find that the top ten important features are TF3, AF5, AF4, NF17, NF11, NF16, NF12, NF14, TF6, and TF1, illustrating that the basic features also play an important role in the detection process.

### D. Parameter Sensitivity Analysis

Next, we provide a sensitivity analysis for the time window parameter  $\delta$  and the probability threshold  $\varepsilon$  to understand their impacts on the performance of the proposed model.

TABLE VII  
PARAMETER ANALYSIS OF PROBABILITY THRESHOLD  $\varepsilon$

Dataset	Metric	0.5	0.6	0.7	0.8	0.9
2014	TPR	<b>0.9318</b>	0.9165	0.8962	0.8653	0.8016
	FPR	0.0535	0.0334	0.0191	0.0093	<b>0.0028</b>
2015	TPR	<b>0.9339</b>	0.9149	0.8863	0.8420	0.7652
	FPR	0.0686	0.0379	0.0182	0.0083	<b>0.0032</b>
2016	TPR	<b>0.9438</b>	0.9318	0.9184	0.8990	0.8510
	FPR	0.0502	0.0356	0.0221	0.0114	<b>0.0037</b>

TABLE VIII  
PERFORMANCE COMPARISON OF DIFFERENT METHODS ON CURRENT TRANSACTION DATA (WITH STANDARD DEVIATION)

Dataset	Method	TPR	FPR	G-Mean
2020 <sup>1</sup>	OCSVM	0.6474±0.0974	<b>0.0572±0.0162</b>	0.7788±0.0567
	IF	<b>0.8537±0.0774</b>	0.5431±0.0539	0.6220±0.0350
	LR	0.0±0.0	0.0±0.0*	0.0±0.0
	DT	0.0685±0.0599	0.0±0.0*	0.2236±0.1368
	IS1	0.8189±0.0629	0.4121±0.0007	0.6933±0.0268
	IS2	0.8189±0.0629	0.4006±0.0008	0.7001±0.0270
	Our method	0.7874±0.0671	0.0991±0.0104	<b>0.8413±0.0327</b>

<sup>1</sup> This dataset contains 1,500,000 consecutive transactions since Mar. 25, 2020.

\* The marked FPRs imply that there exist overfitting problems in LR and DT as the positive instances are more likely to be predicted as negative instances.

Fig. 8 shows the results in terms of TPR, FPR, and G-Mean of our model versus time window  $\delta \in \{0.25, 0.5, 1, 2, \dots, 6\}$  h. We can observe that the curves of the metrics are generally stable, which illustrates that our model can steadily obtain relatively good results under different settings of parameter  $\delta$  in the testing domain.

We also provide the TPR and FPR results of our model under different probability threshold  $\varepsilon$  for voting in Table VII. We can observe that the lower  $\varepsilon$  is, the higher TPR is. While for FPR, it becomes lower with a larger  $\varepsilon$ . For practical applications, we can choose an appropriate threshold according to our specific requirement of pursuing a higher TPR or ensuring a lower FPR.

### E. Discussion

In recent years, some new techniques, such as segregated witness (SegWit) [21] and lightning network (LN) [22], have

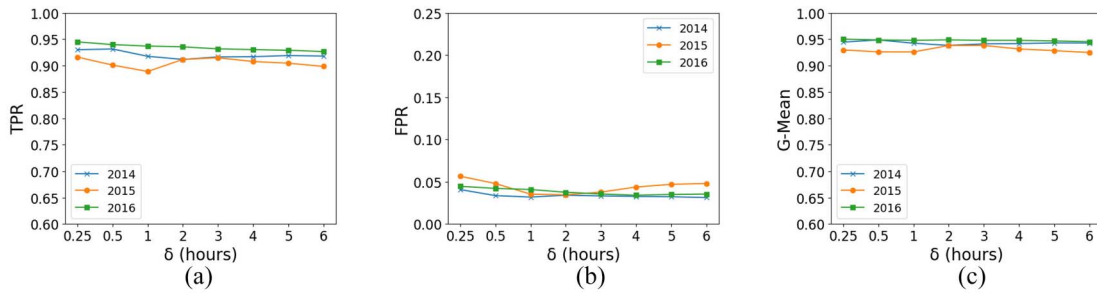


Fig. 8. Parameter analysis of time window  $\delta$ . (a)–(c) Results of TPR, FPR, and G-Mean measured against different  $\delta$ , respectively.

been developed in the Bitcoin community, bringing some new changes to Bitcoin transactions and addresses. The SegWit soft-fork accepted in August 2017 is a solution for the transaction malleability problem [23] by redesigning the transaction structure and segregating the witness data so that modifications to the witness would not change the transaction hash. In addition, the use of SegWit can reduce the size of a transaction and increase the number of transactions contained in a block. LN is an off-chain solution to improve the scalability of Bitcoin. For two Bitcoin users, they can open an LN channel by locking some Bitcoins in a 2-of-2 multisignature address through an on-chain transaction, and then they can trade with each other via this channel without recording in Bitcoin. Once they broadcast a commitment transaction onto the blockchain to get their respective balance from the multisignature address, the LN channel will be closed. Different from traditional transactions in Bitcoin, the use of LN only results in two transaction records each time for the opening and closing of a channel. Besides, almost all the LN transactions are based on 2-of-2 multisignature addresses in witness scripts after the activation of SegWit, and these addresses are native Segwit addresses started with “bc.”

Since WalletExplorer does not include the new emerging services after 2016, we have conducted experiments with three snapshots during 2014–2016. To justify how the data in our experiments are relevant for current transactions and addresses, we collect 1 500 000 consecutive transactions on Bitcoin since March 25, 2020 and examine the performance of our model on the dataset which is referred to as the 2020 dataset in Table VIII. The active labeled addresses of mixing services crawled from WalletExplorer in this snapshot are 89 addresses belonging to Bitcoin Fog, and the number of unlabeled addresses is 1 654 175 after filtering. We also conduct a method comparison experiment on this new dataset under the same experimental settings as Section VI-B. The performance comparison results displayed in Table VIII show that our model still performs best in terms of G-mean. Yet its performance in terms TPR and FPR is slightly worse than the best method. These results may be due to two possible reasons. One is the small amount of labels in the 2020 dataset, and the other is the introduction of some new techniques like LN. To further enhance the detection effectiveness on current transactions and addresses, the most direct method is to collect more labels via using some mixing services and then conduct analysis on them for better capturing their features. Another feasible solution is to utilize link prediction to enrich the link

information of native Segwit addresses, which can help us better identify their ownership.

## VII. RELATED WORK

As a new technology, blockchain has attracted intensive interests of researchers from various fields. Since the transaction data of blockchain systems are publicly accessible, they have been extensively studied to mine some network properties for transaction networks [24]–[26], to cluster addresses sharing the same ownership [27], [28], and to discover some specific activities, such as scams [29]–[31], attacks [32], dark market trading [33], etc. Chen *et al.* [34] conducted a graph analysis and abnormal contract detection on Ethereum with a money flow graph, smart contract creation graph, and smart contract invocation graph. Tam *et al.* [35] proposed a graph convolution network (GCN)-based embedding method to identify illicit accounts within the e-payment networks including the Ethereum transaction network. In [36], typical abnormal transaction patterns for Bitcoin market manipulation were mined by inspecting the base networks with singular value decomposition metrics.

For the issue of money laundering detection in Bitcoin, Möser *et al.* [5] provided an inquiry into the operation models of three mixing services and tried to trace the anonymous transactions. Weber *et al.* [37] emphasized the importance of anti-money laundering (AML) regulations in the financial system, and contributed the Elliptic dataset for illicit activity detection in Bitcoin. Ranshous *et al.* [38] introduced the idea of motifs in directed hypergraphs and recognized some specific laundering patterns for Bitcoin exchanges. Bitcoveview, a visualization tool for Bitcoin, was proposed to visualize how and when an address mixes its money [39]. Recently, a cryptocurrency exchange platform called ShapeShift<sup>4</sup> was reported to be involved in money laundering activities by moving Bitcoins to other privacy-enhancing cryptocurrencies, such as Zcash [40] and Monero [41]. To address this problem, Yousaf *et al.* [42] proposed recognition methods for tracing cross-ledger transaction behaviors. Yet these techniques do not focus on identifying addresses enrolling in mixing. Another work shed light on the problem of Bitcoin mixing detection and tackled it as a community outlier detection problem [8]. However, this work is in lack of generalization for different mixing services and it only utilizes the topology information of the transaction network. Inspired by a related study about

<sup>4</sup><https://classic.shapeshift.com/>

detecting Ponzi schemes on Ethereum [30], in this article, we propose features from multilevel, trying to discover the transaction patterns of mixing services for the enhancement of the generalization ability.

It is worth mentioning that the network motifs we used, which are defined as the recurrent subgraph patterns of complex networks [10], play an important role in characterizing the behavior of mixing services. As the simple building blocks in complex systems, motifs have been demonstrated as a powerful tool for revealing higher-order organizations [43] and functional properties. Since many interactions between objects are intermittent rather than persistent, network motifs combined with temporal information were proposed to characterize dynamic homogeneous network [11], and also had an extensive version in HIN [44]. Recently, there are many studies utilized network motifs in blockchain transaction network mining tasks, such as price prediction [45], [46], network property analysis [47], exchange pattern mining [38], and so on. Network attributes play important roles in network mining tasks [12], nevertheless, most of these network mining studies fail to consider the rich information of network attributes when characterizing the interaction patterns with motifs.

#### VIII. CONCLUSION AND FUTURE WORK

In this work, we studied the Bitcoin mixing detection problem and conducted a systematic analysis to characterize how addresses belonging to mixing services behave in the Bitcoin transaction network. To mine the dynamic process and transaction patterns in Bitcoin more comprehensively, we employed the Bitcoin transaction records to build two temporal directed graphs, including a homogeneous AAIN and a heterogeneous TAIN. For TAIN, we proposed a novel concept of ATH motifs to integrate edge attribute information with higher-order structures. We developed hybrid motifs, including temporal motifs in AAIN and ATH motifs in TAIN, as the key features for mixing detection. With several designed features, we built a PU learning-based detection model to handle the issue of extremely label imbalance of the mixing detection problem. Extensive experimental results on three real Bitcoin datasets demonstrated the effectiveness of our detection model.

This work revealed some critical transaction behaviors which can distinguish the addresses belonging to mixing services, and then designed an effective method to detect these addresses. One concern is that the mixing service providers may update their mechanisms to eliminate these typical behaviors and avoid being detected. For example, they can inject extra Bitcoins from external addresses and those injected tainted Bitcoins may sit in their addresses for a long time to fake the flow of Bitcoins, or they may increase and randomize the interval between the arrival and departure of Bitcoins, to avoid creating the discussed motifs within a specific time window. Since the available data are intrinsically mostly unlabeled and our detection model is based on the prior information, these unknown complex mixing strategies may exist and may not be detected. For future work, we will look for more adaptive strategies to defend these updated privacy-enhancing

techniques, such as applying link prediction to enrich the money flow information.

#### REFERENCES

- [1] S. Nakamoto. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. [Online]. Available: <https://bitcoin.org/bitcoin.pdf>
- [2] Y. Yuan and F.-Y. Wang, "Blockchain and cryptocurrencies: Model, techniques, and applications," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 9, pp. 1421–1428, Sep. 2018.
- [3] W. Chen and Z. Zheng, "Blockchain data analysis: A review of status, trends and challenges," *J. Comput. Res. Devices*, vol. 55, no. 4, pp. 1853–1870, 2018.
- [4] A. Murko and S. L. R. Vrhovec, "Bitcoin adoption: Scams and anonymity May not matter but trust into bitcoin security does," in *Proc. 3rd Central Eur. Cybersecurity Conf. (CECC)*, New York, NY, USA, 2019, pp. 1–6.
- [5] M. Möser, R. Böhme, and D. Breuker, "An inquiry into money laundering tools in the bitcoin ecosystem," in *Proc. APWG eCrime Res. Summit (eCRS)*, San Francisco, CA, USA, 2013, pp. 1–14.
- [6] R. van Wegberg, J.-J. Oerlemans, and O. van Deventer, "Bitcoin money laundering: Mixed results? An explorative study on money laundering of cybercrime proceeds using bitcoin," *J. Financ. Crime*, vol. 25, no. 2, pp. 419–435, 2018.
- [7] Y. J. Fanusie and T. Robinson. (2018). *Bitcoin Laundering: An Analysis of Illicit Flows Into Digital Currency Services*. [Online]. Available: [https://www.fdd.org/wp-content/uploads/2018/01/MEMO\\_Bitcoin\\_Laundering.pdf](https://www.fdd.org/wp-content/uploads/2018/01/MEMO_Bitcoin_Laundering.pdf)
- [8] M. A. Prado-Romero, C. Doerr, and A. Gago-Alonso, "Discovering bitcoin mixing using anomaly detection," in *Proc. Iberoamer. Congr. Pattern Recognit. (CIARP)*, 2017, pp. 534–541.
- [9] B. Liu, Y. Dai, X. Li, W. S. Lee, and P. S. Yu, "Building text classifiers using positive and unlabeled examples," in *Proc. 3rd IEEE Int. Conf. Data Min. (ICDM)*, vol. 3, 2003, pp. 179–188.
- [10] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon, "Network motifs: Simple building blocks of complex networks," *Science*, vol. 298, no. 5594, pp. 824–827, 2002.
- [11] A. Paranjape, A. R. Benson, and J. Leskovec, "Motifs in temporal networks," in *Proc. 10th ACM Int. Conf. Web Search Data Min. (WSDM)*, Cambridge, U.K., 2017, pp. 601–610.
- [12] P.-Z. Li, L. Huang, C.-D. Wang, D. Huang, and J.-H. Lai, "Community detection using attribute homogenous motif," *IEEE Access*, vol. 6, pp. 47707–47716, 2018.
- [13] T. de Balthasar and J. Hernandez-Castro, "An analysis of bitcoin laundry services," in *Proc. Nordic Conf. Secure IT Syst. (NordSec)*, 2017, pp. 297–312.
- [14] E. Wong, B. Baur, S. Quader, and C.-H. Huang, "Biological network motif detection: Principles and practice," *Briefings Bioinform.*, vol. 13, no. 2, pp. 202–215, 2012.
- [15] M. E. J. Newman, *Networks: An Introduction*. Oxford, U.K.: Oxford Univ. Press, 2010.
- [16] M. Möser, R. Böhme, and D. Breuker, "Towards risk scoring of bitcoin transactions," in *Proc. Int. Conf. Financ. Cryptogr. Data Security (FC)*, 2014, pp. 16–32.
- [17] M. Kubat and S. Matwin, "Addressing the curse of imbalanced training sets: One-sided selection," in *Proc. Int. Conf. Mach. Learn. (ICML)*, vol. 97, Nashville, TN, USA, 1997, pp. 179–186.
- [18] Y. Tang, Y.-Q. Zhang, N. V. Chawla, and S. Krasser, "SVMs modeling for highly imbalanced classification," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 1, pp. 281–288, Feb. 2009.
- [19] B. Tang, H. He, P. M. Baggenstoss, and S. Kay, "A Bayesian classification approach using class-specific features for text categorization," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 6, pp. 1602–1606, Jun. 2016.
- [20] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proc. 8th IEEE Int. Conf. Data Min. (ICDM)*, Pisa, Italy, 2008, pp. 413–422.
- [21] E. Lombrozo, J. Lau, and P. Wuille. (2015). *BIP141: Segregated Witness (Consensus Layer)*. [Online]. Available: <https://github.com/bitcoin/bips/blob/master/bip-0141.mediawiki>
- [22] J. Poon and T. Dryja. (2016). *The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments*. [Online]. Available: <https://www.bitcoinlightning.com/wp-content/uploads/2018/03/lightning-network-paper.pdf>

- [23] C. Decker and R. Wattenhofer, "Bitcoin transaction malleability and MtGox," in *Proc. Eur. Symp. Res. Comput. Security (ESORICS)*, 2014, pp. 313–326.
- [24] F. Reid and M. Harrigan, *An Analysis of Anonymity in the Bitcoin System*. New York, NY, USA: Springer, 2013, pp. 197–223.
- [25] I. Alqassem, I. Rahwan, and D. Svetinovic, "The anti-social system properties: Bitcoin network data analysis," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 1, pp. 21–31, Jan. 2020.
- [26] D. Ron and A. Shamir, "Quantitative analysis of the full bitcoin transaction graph," in *Proc. Int. Conf. Financ. Cryptogr. Data Security (FC)*, 2013, pp. 6–24.
- [27] T.-H. Chang and D. Svetinovic, "Improving bitcoin ownership identification using transaction patterns analysis," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 1, pp. 9–20, Jan. 2020.
- [28] E. Androulaki, G. Karame, M. Roeschlin, T. Scherer, and S. Capkun, "Evaluating user privacy in bitcoin," in *Proc. Int. Conf. Financ. Cryptogr. Data Security (FC)*, 2013, pp. 34–51.
- [29] M. Vasek and T. Moore, "There's no free lunch, even using bitcoin: Tracking the popularity and profits of virtual currency scams," in *Proc. Int. Conf. Financ. Cryptogr. Data Security (FC)*, 2015, pp. 44–61.
- [30] W. Chen, Z. Zheng, J. Cui, E. Ngai, P. Zheng, and Y. Zhou, "Detecting Ponzi schemes on Ethereum: Towards healthier blockchain technology," in *Proc. World Wide Web Conf. (WWW)*, Lyon, France, 2018, pp. 1409–1418.
- [31] W. Chen, Z. Zheng, E. C.-H. Ngai, P. Zheng, and Y. Zhou, "Exploiting blockchain data to detect smart Ponzi schemes on Ethereum," *IEEE Access*, vol. 7, pp. 37575–37586, 2019.
- [32] F. Tschorsch and B. Scheuermann, "Bitcoin and beyond: A technical survey on decentralized digital currencies," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 2084–2123, 3rd Quart., 2016.
- [33] N. Christin, "Traveling the silk road: A measurement analysis of a large anonymous online marketplace," in *Proc. 22nd Int. Conf. World Wide Web (WWW)*, Rio de Janeiro, Brazil, 2013, pp. 213–224.
- [34] T. Chen *et al.*, "Understanding Ethereum via graph analysis," in *Proc. INFOCOM Conf. Comput. Commun.*, Honolulu, HI, USA, 2018, pp. 1484–1492.
- [35] D. S. H. Tam, W. C. Lau, B. Hu, Q. F. Ying, D. M. Chiu, and H. Liu, "Identifying illicit accounts in large scale e-payment networks—A graph representation learning approach," 2019. [Online]. Available: <http://arxiv.org/abs/1906.05546>.
- [36] W. Chen, J. Wu, Z. Zheng, C. Chen, and Y. Zhou, "Market manipulation of bitcoin: Evidence from mining the Mt. Gox transaction network," in *Proc. INFOCOM Conf. Comput. Commun.*, Paris, France, 2019, pp. 964–972.
- [37] M. Weber *et al.*, "Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics," 2019. [Online]. Available: <http://arxiv.org/abs/1908.02591>.
- [38] S. Ranshous *et al.*, "Exchange pattern mining in the bitcoin transaction directed hypergraph," in *Proc. Int. Conf. Financ. Cryptogr. Data Security (FC)*, 2017, pp. 248–263.
- [39] G. D. Battista, V. D. Donato, M. Patrignani, M. Pizzonia, V. Roselli, and R. Tamassia, "Bitconeview: Visualization of flows in the bitcoin transaction graph," in *Proc. IEEE Symp. Vis. Cyber Security (VizSec)*, Chicago, IL, USA, 2015, pp. 1–8.
- [40] G. Kappos, H. Yousaf, M. Maller, and S. Meiklejohn, "An empirical analysis of anonymity in Zcash," in *Proc. 27th USENIX Conf. Security Symp.*, Baltimore, MD, USA, 2018, pp. 463–477.
- [41] M. Möser *et al.*, "An empirical analysis of traceability in the Monero blockchain," in *Proc. Privacy Enhancing Technol.*, vol. 2018, no. 3, pp. 143–163, 2018.
- [42] H. Yousaf, G. Kappos, and S. Meiklejohn, "Tracing transactions across cryptocurrency ledgers," in *Proc. 28th USENIX Conf. Security Symp.*, Santa Clara, CA, USA, 2019, pp. 837–850.
- [43] A. R. Benson, D. F. Gleich, and J. Leskovec, "Higher-order organization of complex networks," *Science*, vol. 353, no. 6295, pp. 163–166, 2016.
- [44] Y. Li, Z. Lou, Y. Shi, and J. Han, "Temporal motifs in heterogeneous information networks," in *Proc. MLG Workshop*, London, U.K., 2018. [Online]. Available: <http://www.mlgworkshop.org/2018/>
- [45] C. G. Akcora, A. K. Dey, Y. R. Gel, and M. Kantarcioglu, "Forecasting bitcoin price with graph chainlets," in *Proc. Pac.-Asia Conf. Knowl. Discov. Data Min. (PAKDD)*, 2018, pp. 765–776.
- [46] N. C. Abay *et al.*, "ChainNet: Learning on blockchain graphs with topological features," 2019. [Online]. Available: <http://arxiv.org/abs/1908.06971>.
- [47] P. Moreno-Sanchez, N. Modi, R. Songhela, A. Kate, and S. Fahmy, "Mind your credit: Assessing the health of the Ripple credit network," in *Proc. World Wide Web Conf. (WWW)*, Lyon, France, 2018, pp. 329–338.



**Jiajing Wu** (Senior Member, IEEE) received the Ph.D. degree in electronic and information engineering from Hong Kong Polytechnic University, Hong Kong, in 2014.

In 2015, she joined Sun Yat-sen University, Guangzhou, China, where she is currently an Associate Professor. Her research focus includes blockchain, graph mining, and network science.

Dr. Wu was a recipient of the Hong Kong Ph.D. Fellowship Scheme during her Ph.D. study in Hong Kong from 2010 to 2014. She serves as an Associate

Editor for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—PART II: EXPRESS BRIEFS.



**Jieli Liu** received the B.Eng. degree in software engineering from Sun Yat-sen University, Guangzhou, China, in 2019, where she is currently pursuing the M.Sc. degree with the School of Computer Science and Engineering.

Her current research interests include blockchain, network science, data mining, and machine learning with graphs.



**Weili Chen** received the Ph.D. degree in computer science and engineering from the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China, in 2019.

He is currently a Postdoctoral Associate Research Fellow with Sun Yat-sen University. His research interests include blockchain, data mining, and machine learning.



**Huawei Huang** (Member, IEEE) received the Ph.D. degree in computer science and engineering from the University of Aizu, Aizuwakamatsu, Japan, in 2016.

He is currently an Associate Professor with Sun Yat-sen University, Guangzhou, China. He has served as a Research Fellow with JSPS, Tokyo, Japan, from 2016 to 2018, a Visiting Scholar with Hong Kong Polytechnic University, Hong Kong, from 2017 to 2018, and an Assistant Professor with Kyoto University, Kyoto, Japan, from 2018 to 2019. His research interests include blockchain and intel-

ligent computing.

Dr. Huang received the Best Paper Award from TrustCom2016. He is a member of ACM.



**Zibin Zheng** (Senior Member, IEEE) received the Ph.D. degree in computer science and engineering from the Chinese University of Hong Kong, Hong Kong, in 2011.

He is currently a Professor of Data and Computer Science with Sun Yat-sen University, Guangzhou, China. He serves as the Chair of the Software Engineering Department, Pearl River Young Scholars, and the Founding Chair of the Services Society Young Scientists Forum. In the past five years, he published over 120 international journal and conference papers, including three ESI highly cited papers and 40 ACM/IEEE TRANSACTIONS papers. According to Google Scholar, his papers have more than 6300 citations, with an H-index of 41. His research interests include blockchain, services computing, software engineering, and financial big data.

Dr. Zheng was a recipient of several awards, including the Outstanding Thesis Award of CUHK in 2012, the ACM SIGSOFT Distinguished Paper Award at ICSE2010, the Best Student Paper Award at ICWS2010, and the IBM Ph.D. Fellowship Award. He served as the CollaborateCom'16 General Co-Chair, the ICIOT'18 PC Co-Chair, and the IoV'14 PC Co-Chair.



**Yan Zhang** (Fellow, IEEE) received the Ph.D. degree in electrical and electronics engineering from the School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore, in 2005.

He is currently a Full Professor with the Department of Informatics, University of Oslo, Oslo, Norway. His research interests include next-generation wireless networks leading to 5G beyond/6G, and green and secure cyber-physical systems (e.g., smart grid and transport).

Prof. Zhang was a recipient of the Global Highly Cited Researcher Award (Web of Science top 1% most cited worldwide) in 2018 and 2019. He is an Editor (or Area Editor and Associate Editor) for several IEEE publications, including *IEEE Communications Magazine*, *IEEE Network Magazine*, *IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING*, *IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY*, *IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS*, *IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING*, *IEEE COMMUNICATIONS SURVEY AND TUTORIALS*, *IEEE INTERNET OF THINGS JOURNAL*, *IEEE SYSTEMS JOURNAL*, *IEEE Vehicular Technology Magazine*, and *IEEE BLOCKCHAIN TECHNICAL BRIEFS*. He is a Symposium/Track Chair in a number of conferences, including IEEE ICC 2021, IEEE Globecom 2017, IEEE PIMRC 2016, and IEEE SmartGridComm 2015. He is an IEEE Vehicular Technology Society Distinguished Lecturer from 2016 to 2020 and he is named as the CCF 2019 Distinguished Speaker. He is the Chair of the IEEE Communications Society Technical Committee on Green Communications and Computing. He is an Elected Member of the CCF Technical Committee of Blockchain. He is a Fellow of IET and an Elected Member of Academia Europaea and the Norwegian Academy of Technological Sciences (NTVA).