Detecting Mixing Services via Mining Bitcoin Transaction Network with Hybrid Motifs

Jiajing Wu, *Member, IEEE,* Jieli Liu, Weili Chen, Huawei Huang, *Member, IEEE,* Zibin Zheng, *Senior Member, IEEE,* and Yan Zhang, *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 laundry to complicate trailing illicit fund. In this paper, 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*—Bitcoin, mixing services, network motifs, network mining, anti-money laundering.**

I. Introduction

BITCOIN, 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, thus triggering the emergence of various illegal activities in the Bitcoin ecosystem [3]. For instance, about 7,000 Bitcoins which worth $41 million have been 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. This way may easily expose the identity of the thieves via the identity information provided by the exchanges. Thus the stolen Bitcoins usually need to be laundered into "clean" Bitcoins by some techniques before they are cashed out. It has been demonstrated that, mixing services such as BitLaundry, Helix Light, Bitcoin Fog, etc., have involved in this process of *money laundry* [5] and can be regarded as significant tools for concealing illicit profits in Bitcoin.

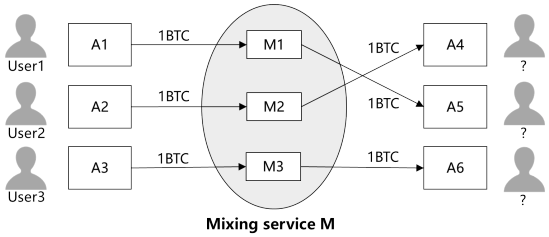


Fig. 1. An example of mixing services, which can conceal the identity of users and complicate fund tracing by participating in a transaction with multiple users.

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 1 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 select 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. Meanwhile, due to the pseudonymous requirements of Bitcoin, it is unlikely to enforce Know-Your-Customer (KYC) processes, which are guidelines in anti-money laundering. Therefore, the study on 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 paper, 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. Fanusie et al. [6] pointed out that mixing services and exchanges are two key components in laundering Bitcoins while mixing services

Manuscript received January xx, 2020. This work was supported in part by the National Natural Science Foundation of China under Grants 61973325 and 61503420, and in part by the Fundamental Research Funds for the Central Universities under Grant 17lgpy120.

J. Wu, J. Liu, W. Chen, H. Huang and Z. Zheng are with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, China. (Email: wujiajing@mail.sysu.edu.cn)

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

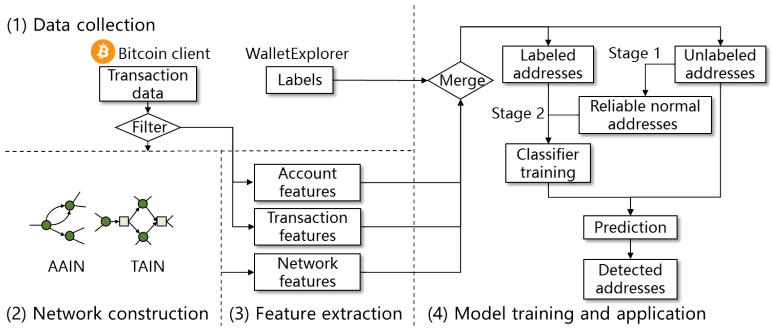


Fig. 2. An overview of the proposed Bitcoin mixing detection framework including four modules, namely, data collection, network construction, feature extraction, model training and application.

have a higher propensity to be used in laundering illicit money. To answer how mixing services work, the operation model of five mixing services were studied by reverse-engineering methods in [5]. Based on the observations given by [5], Prado-Romero proposed the problem of mixing detection and tackled this problem by exploiting the method of community outlier detection [7]. 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 evasion techniques and transaction patterns.** Mixing services are provided by different third-party platforms, and their evasion techniques and transaction 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 [8] 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 [9]–[11]. 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 vital features for the detection of mixing services.

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 craws the label information from WalletExplorer[1]. (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. (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 paper 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 and a low false positive rate in Bitcoin mixing detection, which facilitates fund tracing and crime detection in the Bitcoin ecosystem.

The remaining sections of this paper are organized as follows. Sections II to 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 paper in Section VIII.

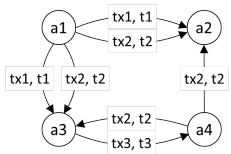
II. Data Collection

The transaction data of Bitcoin are contained in blocks orderly, and are publicly accessible by running a Bitcoin client. WalletExplorer is a smart Bitcoin block explorer providing name information of addresses by registering to some services, making transactions with them and observing how the Bitcoin flows merge, and the name database of WalletExplorer in not updated since 2016. From the name dataset provided by WalletExplorer, we crawl the labeled addresses belonging to mixing services recorded between 2014 and 2016. As Bitcoin transactions increase explosively during this period,

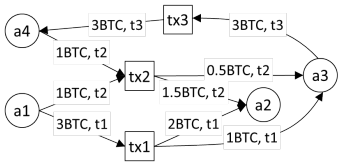
[1]https://www.walletexplorer.com.

TxID	Time	Address	Value	Type
tx1	t1	a1	3BTC	input
		a2	2BTC	output
		a3	1BTC	output
tx2	t2	a1	1BTC	input
		a4	1BTC	input
		a2	1.5BTC	output
		a3	0.5BTC	output
tx3	t3	a3	3BTC	input
		a4	3BTC	output

(a) Transaction data.



(b) AAIN.



(c) TAIN.

Fig. 3. An example for our network construction where (a) is the raw transaction data, (b) and (c) are the corresponding Address-Address Interaction Network (AAIN) and Transaction-Address Interaction Network (TAIN) respectively. Here $t1 < t2 < t3$, indicating that $tx1$ happens earlier than $tx2$, and $tx2$ happens earlier than $tx3$.

TABLE I
STATISTICS OF THE DATASETS.

Dataset	Start time	Unlabeled address	Labeled address[1]		
			BitcoinFog	BitLaunder	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

[1] Addresses of three mixing services including Bitcoin Fog, BitLaunder and Helix Mixer crawled from WalletExplorer are as our labeled addresses.

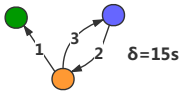
we consider three snapshots of Bitcoin transaction data with mixing labels between November 2014 and January 2016 with six months being the sampling interval. Each snapshot contains 1,500,000 transaction records. The three snapshots are referred to as the 2014, 2015 and 2016 datasets. Table I shows the 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.

As mixing services serve as intermediaries to anonymize the transactions, we filter addresses with either only input transactions or output transactions which obviously not belong to mixing services. By applying this simple rule, 131 labeled addresses and 1,635,904 unlabeled addresses are filtered from the three datasets, which occupy 0.9% and 22.2% of their corresponding class, respectively.

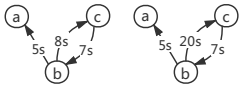
III. Network Construction and Motif Definition

Transaction records of Bitcoin can be abstracted as a huge network, where each node refers to a Bitcoin address and each edge represents a transaction process between addresses. Network motifs, which can be regarded as small subgraph patterns in networks, have been demonstrated as important tools for characterizing higher-order interactions and understanding various properties of complex systems [9], [10].

We first construct a homogeneous Address-Address Interaction Network (AAIN), which is a temporal directed network, to investigate the interaction patterns of addresses.



(a) A 3-node, 3-edge, δ-temporal motif M.



(b) Two motif instances of M.

(c) Two graph patterns which are not instances of M.

Fig. 4. An example of a motif (a) and its instances (b). 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 δ is set as 15s).

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 Transaction-Address Interaction Network (TAIN) 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 sent to or received from a transaction. Therefore, compared with AAIN, TAIN can display the strength of money transfer more clearly. Fig. 3 gives an example for the construction of AAIN and TAIN. In the following, we present the definition of AAIN, TAIN and their motifs in detail.

A. AAIN and Temporal Motifs

Definition 1 (AAIN): An Address-Address Interaction Network (AAIN) is a temporal network $G = (V, E)$, 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.

AAIN is a temporal direct multigraph which can reflect the flowing directions of money. As shown in Fig. 3(b), Bitcoins of $a1$ can be transferred to $a4$ via $tx1$, $tx2$ and $tx3$ while Bitcoins in $a3$ cannot reach $a2$ because $tx2$ occurs ahead of $tx3$.

Definition 2 (Temporal Motifs): Temporal motifs are defined as interconnection patterns occurring in temporal networks [10]. Particularly, a k-node, l-edge, δ-temporal motif instance $M_k^l(\delta)$ of a temporal network $G = (V, E)$ can be represented as

$$M_k^l(\delta) = (V_M^k, E_M^l, \delta),$$

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 δ is a time window indicating that all of edges in the motif occur within a δ duration, i.e., an increased sequence $t_1, t_2, ..., t_l$ which records the timestamp of each edge in the motif instance satisfies $t_1 \leqslant t_2 \leqslant ... \leqslant t_l$ and $t_l - t_1 \leqslant \delta$.

Different from static network motifs, temporal motifs well preserve the time-ordered sequence of contacts in a time

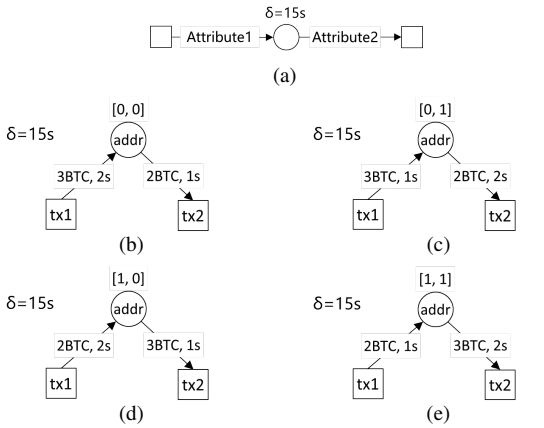


Fig. 5. (b)-(e) are four instances of the ATH motif (a) in transaction network. The edge vector Γ_V, which maps the amount value attribute to the first bit and the time information to the second bit, can differentiate varieties of transaction patterns with the same topology. In this case, the first bit of Γ_V is set as 0 if the amount of $tx1$ is higher than $tx2$, and as 1 otherwise; the second bit of Γ_V is set as 0 if the time of $tx1$ is later than $tx2$, and as 1 otherwise.

window, being effective in analyzing temporal structure of complex networks. Figs. 4(a)-(b) illustrate a 3-node, 3-edge, δ-temporal motif M and its instances, while graph patterns in Fig. 4(c) are not instances of M because their edge order or occurring time window does not satisfy the condition.

B. TAIN and ATH Motifs

An attributed temporal heterogeneous information network (HIN) is defined as a directed graph $G = (V, E, \Omega)$ with $\varphi_V : V \to \Gamma_V$ for node types mapping, $\varphi_E : E \to \Gamma_E$ for edge types mapping satisfying $|\Gamma_V| > 1$ or $|\Gamma_E| > 1$, and Ω denoting the set of attributes attached to edges in the graph. Based on the attributed temporal HIN, we present the definition of TAIN and attributed temporal heterogeneous motifs (ATH motifs).

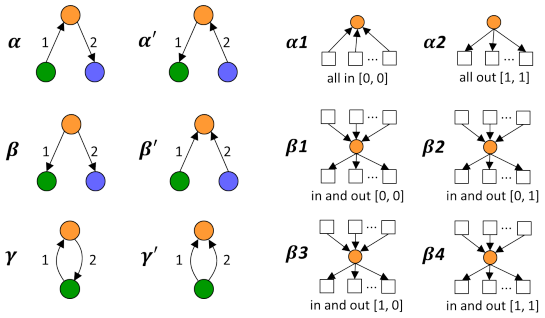
Definition 3 (TAIN): A TAIN is an attributed temporal heterogeneous information network $G = (V, E, \Omega)$, where $\Gamma_V = \{\text{address,transaction}\}$ denotes the set of node types, $\Gamma_E = \{\textit{transaction-address}, \textit{address-transaction}\}$ denotes the set of edge types, and Ω denotes the set of edge attributes including transaction amount and transaction time.

A *transaction-address* edge $(u, tx_{in}, a, t1)$ denotes that an input transaction tx_{in} happens at time $t1$ and transfers a Bitcoins into an address u, while an *address-transaction* edge $(v, tx_{out}, b, t2)$ denotes that an output transaction tx_{out} happens at time $t2$ and transfers b Bitcoins out of an address v. An example of TAIN is illustrated in Fig. 3(c), where the square nodes represent transactions and the circle nodes represent addresses.

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

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

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



(a) 2-edge δ-temporal motifs. (b) Attributed δ-ATH motifs.

Fig. 6. Hybrid motifs in Bitcoin, including temporal motifs in AAIN (a) and ATH motifs in TAIN (b). The weight of edges in temporal motifs represents the chronological order and each ATH motif accompanies with a mapped vector.

Fig. 5 shows us an example of ATH motif and its four instances in a transaction network. Though the topology of these motif instances are the same, the attribute information separates these instances into four different transaction patterns. For example, the instance in Fig. 5(c) represents the transaction pattern of receiving money first and sending out less money later, while the instance in Fig. 5(d) stands for sending money out firstly and receiving less money later, which has an opposite transaction order and leads to a negative balance.

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.

For AAIN, we consider six simplest transaction patterns with two edges and represent them as six kinds of 2-edge δ-temporal motifs shown in Fig. 6(a). These six kinds of motifs illustrate how an address interacts with other addresses within a δ duration. For example, the α motif represents that an address first receives money from a neighbor and then transfers money to another neighbor while the α' motif represents an opposite transaction order.

For TAIN, we abstract the transactions of an address occurring within a δ time window as three kinds of topological

structures: *all in*, *all out*, *in and out*, which illustrate only input transactions, only output transactions, both input and output transactions occurring within the time window, respectively. By taking the amount attribute 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 transaction patterns can be represented as ATH motifs in TAIN. As shown in Fig. 6(b), each kind of ATH motif has a binary value function defining each bit of Γ_Ω as follows.

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

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, $\bar{t}_{in}$ 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 Γ_Ω is defined as $[0, 0]$. Contrarily, Γ_Ω of *all out* patterns are given as $[1, 1]$.

We calculate the proportion of each kind of motifs described above. 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-N6: The proportion of each kind of 2-edge δ-temporal motifs given in Fig. 6(a).

NF7-N12: The proportion of each kind of ATH motifs given in Fig. 6(b).

NF13: Value of in-degree.

NF14: Value of out-degree.

NF15: Average number of co-input brothers.

NF16: Average number of co-output brothers.

NF17: Number of unique successors.

NF18: Number of unique predecessors.

Tables II and III describe the average fraction of each kind of temporal motifs and ATH motifs with $\delta = 3$ hours, respectively. The interpretation of this selected time window will be provided in Section IV-C. Based on the statistics, we summarize several findings as follows:

- **Finding 1.** The average fraction of α pattern is much higher than the fraction of α' pattern, and the fraction of $\beta 2$ pattern far outstrips the other kinds of *in and out* motifs. 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 firstly and sending money out latter with a balance not less than 0.
- **Finding 2.** Based on the results of γ and γ' patterns, we can conclude that non-mixing service entity may reuse some addresses in a short time while this situation seldom happens for mixing services.
- **Finding 3.** By comparing the fraction of β with that of β' and the fraction of $\alpha 1$ with that of $\alpha 2$, we can see that mixing services prefer dispersing the "dirty" Bitcoins to others, which is a usually adopted strategy for Bitcoin mixing.

TABLE II
AVERAGE FRACTION OF δ-TEMPORAL MOTIFS ($\delta = 3$ HOURS).

Temporal motifs	α	α'	β	β'	γ	γ'
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

TABLE III
AVERAGE FRACTION OF δ-ATH MOTIFS ($\delta = 3$ HOURS).

ATH motifs	$\alpha 1$	$\alpha 2$	$\beta 1$	$\beta 2$	$\beta 3$	$\beta 4$
Labeled address	0.0390	0.4912	0.0037	0.4498	0.0046	0.0117
Unlabeled address	0.1670	0.4932	0.0022	0.3122	0.0051	0.0203

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

AF1: Balance of the address[2].

AF2: Number of input transactions in the snapshot.

AF3: Number of output transactions in the snapshot.

AF4: Total amount of input transactions in the snapshot.

AF5: Total amount of output transactions in the snapshot.

AF6: Amount ratio of total input transactions to total output transactions.

Table IV summarizes the mean, standard deviation (StdDev) and median values of the account features for all the labeled addresses and unlabeled addresses in our datasets. Some notable results can be obtained from Table IV 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

[2]The balance information is crawled from the api of blockchain.info on June 17th, 2019.

TABLE IV
STATISTICS OF ACCOUNT FEATURES.

	AF1	AF2	AF3	AF4	AF5	AF6
Labeled address						
Mean	1.80e-05	1.28	1.28	1.48	1.48	1.00
StdDev	2.01e-03	2.56	2.54	7.22	7.21	0.45
Median	0.00	1.00	1.00	0.32	0.32	1.00
Unlabeled address						
Mean	0.02	1.79	1.87	6.79	6.80	57.50
StdDev	18.64	42.93	43.07	1228.91	1227.68	68766.30
Median	0.00	1.00	1.00	0.11	0.11	1.00

is relatively smaller, as illustrated by a relatively low value of standard deviation.

- **Finding 5.** The average balance value of most labeled addresses is close to 0. For unlabeled addresses, the mean value and standard deviation of the account balance is much larger.
- **Finding 6.** 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.

These observations can fully illustrate that addresses belonging to mixing services act like intermediaries by sending out what they have received in short time.

C. Transaction Features

Next, 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 send out an approximate equal amount (minus the charges for mixing service, and in this paper we do not consider them) 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.

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. 7 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. Two transaction features (referred as TFs) are extracted as follows:

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

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

The cumulative proportion line chart of $\bar{T}$ is shown in Fig. 8. We can observe that the avereage time interval of transaction cycles of mixing services is mostly within 3 hours, 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 relatively short time as they are user-oriented services. For example, Helix[3] has a maximum process time of

[3]https://helixlightgrams.com/

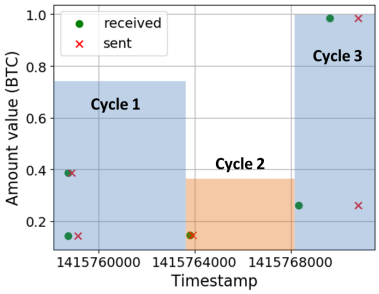


Fig. 7. A transaction cycle is composed of a continuous input stream and a continuous output stream. Three transaction cycles of labeled address "1NsNkSxyYjB9o3QkPT2RjTXST4nGRtfMzS" are shown in this figure.

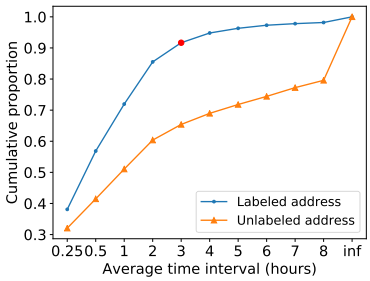


Fig. 8. The cumulative proportion of average time $\bar{T}$.

2 hours to launder Bitcoins. As a result, we preliminarily set $\delta = 3$ hours as the time window of motifs when conducting descriptive statistics and mixing detection.

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.

In stage one, according to the spy technique proposed in [8], we sample a set of spy instances from positive instances with a default sample rate 15%. The rest of the positive instances are 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 the classifier. Here we employ logistic regression as the classifier to ensure efficiency.

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 θ based on the prediction probabilities of the spy instances. The reliable negative instances are selected out according to θ and the prediction probabilities of the unlabeled instances.

We select the probability threshold θ as the value which can maximize the difference of the growth rates between the cumulative proportion of unlabeled instances and spy instances. For each instance i in the instance set, its probability of being predicted as a positive instance is denoted x_i and

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

Dataset	Metric	OCSVM	IF	LR	DT	IS1[1]	IS2[2]	Our method
2014	TPR	0.9205±0.0004	**0.9326±0.0021**	0.1720±0.0018	0.7150±0.0058	0.8208±0.0	0.8208 ±0.0	0.9189±0.0004
	FPR	0.1809±0.0005	0.2082±0.0283	0.0±0.0*	0.0±0.0*	0.0406±0.0002	0.0364±0.0001	**0.0302±0.0004**
	G-Mean	0.8683±0.0003	0.8592±0.0149	0.4147±0.0021	0.8456±0.0034	0.8874±0.0	0.8894±0.0	**0.9440±0.0003**
2015	TPR	0.9026±0.0009	0.9054±0.0034	0.0609±0.0014	0.5979±0.0059	0.7871±0.0	0.7871±0.0	**0.9147±0.0007**
	FPR	0.1672±0.0006	0.2092±0.0259	0.0±0.0*	0.0±0.0*	0.0440±0.0002	0.0449±0.0002	**0.0337±0.0005**
	G-Mean	0.8670±0.0005	0.8460±0.0136	0.2468±0.0028	0.7732±0.0038	0.8674±0.0001	0.8670±0.0	**0.9401±0.0005**
2016	TPR	0.9157±0.0003	0.9204±0.0027	0.0025±0.0	0.3938±0.0023	**0.9409±0.0**	**0.9409±0.0**	0.9343±0.0004
	FPR	0.2265±0.0006	0.3804±0.0301	0.0±0.0*	0.0±0.0*	0.0609±0.0003	0.0604±0.0003	**0.0362±0.0003**
	G-Mean	0.8416±0.0004	0.7550±0.0181	0.0503±0.0	0.6275±0.0018	0.9400±0.0002	0.9403±0.0002	**0.9490±0.0002**

[1,2] IS1 and IS2 use two different inter links counting function, namely relative inter links and total inter links respectively [7].
* The marked FPRs imply that there exists overfitting problems in LR and DT as the positive instances are more likely to be predicted as negative instances.

stored in a set X, namely $x_i \in X$, and then the cumulative distribution function $F(\cdot)$ of X is given by

$$F_X(p) = P(X \leqslant p), \tag{2}$$

where $P(X \leqslant p)$ represents the probability that values in X less than or equal to a value p. Then the growth rate of $F_X(p)$ with Δp as the minute change of probability p ($\Delta p = 0.005$ in our model) is denoted as:

$$\Delta F_X(p) = F_X(p) - F_X(p - \Delta p). \tag{3}$$

According to the prediction probabilities of spy instances and unlabeled instances stored in S and U respectively, the threshold θ is calculated as:

$$\theta = \arg\max_{p \in [0+\Delta p, 1]} (\Delta F_U(p) - \Delta F_S(p)). \tag{4}$$

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.

$$C_+ \sum_{y_i=1} l(y_i, f(\boldsymbol{x_i})) + C_- \sum_{y_i=-1} l(y_i, f(\boldsymbol{x_i})) + \lambda R(\boldsymbol{w}), \tag{5}$$

where C_+ and C_- denote the penalty coefficients of positive and reliable negative instances, respectively, $l(y_i, f(\boldsymbol{x_i}))$ is the loss term, $R(\boldsymbol{w})$ is the regularization term and λ is the regularization coefficient. In this work, we apply a biased logistic regression (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 ε 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 ε.

VI. EXPERIMENTAL RESULTS

In this section, we conduct a comprehensive evaluation on the proposed detection framework for Bitcoin mixing services. First, we describe our experiment settings. Then, we present the experimental results of the proposed method in comparison with several baseline methods. Next, the effects of motif-based features and other basic features are compared and summarized. Finally, we demonstrate the robustness of our framework via a parameter sensitivity analysis.

TABLE VI
DETECTION RESULTS.

Dataset	Unlabeled address	Detected address (ratio)
2014	594853	39828 (6.7%)
2015	588921	37639 (6.4%)
2016	556679	47546 (8.5%)

A. Experiment Settings

In all the experiments, we set the time window $\delta = 3$ hours and the probability threshold $\varepsilon = 0.6$. All the reported results are averaged over 10 independent experiments.

Datasets. As mentioned in Section II, we obtain three datasets with transaction data from a Bitcoin client as well as labels from WalletExplorer, and then conduct experiments on these three datasets. After the process of address filtering, we divide each dataset into training set and testing set as follows:

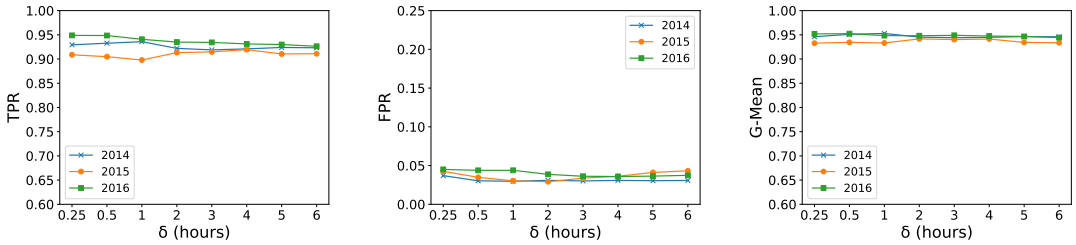
(1) Training set: For stage one, we select 70% unlabeled addresses and 70% labeled addresses to form the training set, and then we can obtain some reliable negative instances. For stage two, the training set is made up of 70% reliable negative instances as well as the labeled addresses used in stage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

Methods for Comparison. Our model is based on PU learning with a two-stage strategy, which is actually a semi-supervised learning method. To evaluate the effectiveness of PU learning in our scenario, we compare our model with a number of baseline methods including one-class support vector machine (OCSVM), isolation forest (IF) [12], logistic regression (LR), decision tree (DT) and InterScore (IS) [7]. 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

TABLE VII
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	0.8917±0.0032	0.8740±0.0	0.8279±0.0003	0.9037±0.0003	0.8946±0.0008	0.8837±0.0056	**0.9189±0.0004**
	FPR	0.1665±0.0050	0.0445±0.0004	0.0782±0.0004	0.0348±0.0001	**0.0285±0.0005**	0.0717±0.0011	0.0302±0.0004
	G-Mean	0.8621±0.0040	0.9138±0.0002	0.8736±0.0001	0.9339±0.0001	0.9323±0.0004	0.9057±0.0032	**0.9440±0.0003**
2015	TPR	0.8091±0.0012	0.8399±0.0	0.8586±0.0	0.8916±0.0004	0.8677±0.0013	0.8531±0.0012	**0.9147±0.0007**
	FPR	0.1420±0.0018	0.1400±0.0006	0.0796±0.0005	0.0652±0.0004	0.0830±0.0031	0.0761±0.0005	**0.0337±0.0005**
	G-Mean	0.8332±0.0011	0.8499±0.0003	0.8889±0.0002	0.9129±0.0002	0.8920±0.0011	0.8878±0.0008	**0.9401±0.0005**
2016	TPR	0.5253±0.0406	0.9300±0.0	0.6736±0.0004	0.9079±0.0005	0.9318±0.0004	0.7985±0.0017	**0.9343±0.0004**
	FPR	0.3658±0.0026	0.0599±0.0003	0.2582±0.0006	0.0398±0.0004	0.0514±0.0003	0.1815±0.0039	**0.0362±0.0003**
	G-Mean	0.5768±0.0212	0.9350±0.0001	0.7069±0.0004	0.9337±0.0002	0.9402±0.0002	0.8084±0.0026	**0.9490±0.0002**

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



(a) TPR measured against different δ. (b) FPR measured against different δ. (c) G-Mean measured against different δ.

Fig. 9. Patameter analysis of time window δ. (a), (b), (c) and (d) show the results of TPR, FPR and G-Mean measured against different δ respectively.

detect mixing service entities containing multiple addresses with community anomaly detection, as addresses belonging to these entities usually have more inter-community 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.

Evaluation metrics. In this work, we evaluate the performance of our model in terms of true positive rate (TPR), false positive rate (FPR) and the Geometric Mean (G-Mean). G-Mean was suggested in [13] and has been widely used as a comprehensive metric in evaluating classification performances on imbalanced datasets [14] [15]. Taking both the accuracy of positive instances and negative instances into account, G-Mean is defined as follows:

$$\text{G}-\text{Mean} = \sqrt{\text{TPR} \times (1 - \text{FPR})} \tag{6}$$

B. Detection Results

Table V compares the performance of our method with the baseline methods. Specifically, the proportion of outliers in the datasets is set to be 8% 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 false positive rate 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 exists 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.

(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 true positive rate exceeding 91% and a low false positive rate below 4% on extremely imbalanced datasets. We then apply our model on unlabeled addresses that not appear in training sets, and the number of detected addresses is given in Table VI. As we can see, about 6%-9% of the unlabeled addresses are detected as suspicious addresses belonging to mixing services in these three snapshots, which facilitates further investigation of illegal transactions on relative users.

C. Validity Proof of Hybrid Motifs

In Section IV, we propose a series of features from the perspective of network, account and transaction levels. According to the results given in Tables II and III, and the analysis in Section IV, we can see that motifs should be critical features for mixing detection. Therefore, we divide all the features

TABLE VIII
PATAMETER ANALYSIS OF PROBABILITY THRESHOLD ε.

Dataset	Metric	0.5	0.6	0.7	0.8	0.9
2014	TPR	**0.9346**	0.9189	0.8970	0.8731	0.8071
	FPR	0.0433	0.0302	0.0201	0.0118	**0.0050**
2015	TPR	**0.9313**	0.9147	0.8957	0.8638	0.8004
	FPR	0.0550	0.0337	0.0189	0.0103	**0.0050**
2016	TPR	**0.9441**	0.9343	0.9207	0.8987	0.8489
	FPR	0.0511	0.0362	0.0232	0.0114	**0.0034**

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 VII 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 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.

In summary, the experimental results demonstrate that hybrid motifs including temporal motif and ATH motifs play an indispensable role in the task of Bitcoin mixing detection.

D. Parameter Sensitivity Analysis

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

Fig. 9 shows the results in terms of TPR, FPR and G-Mean of our model versus time window $\delta \in \{0.25, 0.5, 1, 2, ..., 6\}$ hours. We can observe that the curves of the metrics are generally stable, which illustrates that our model can steadily obtain relative good results under different settings of parameter δ in the testing domain.

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

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 usually publicly accessible, they have been extensively studied and mined to discover some specific activities such as scams [16], gambling [17], dark market trading [18] and so on. Chen et al. conducted a graph analysis and abnormal contract detection on Ethereum with money flow graph, smart contract creation graph and smart contract invocation graph [19]. Tam et al. proposed a Graph Convolution Network (GCN)-based embedding method to identify illicit accounts within the e-payment networks including the Ethereum transaction network [20]. In [21], 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. provided an inquiry into the operation mode of three mixing services and try to trace the anonymous transactions [5]. Weber et al. emphasize the importance of Anti-money laundering (AML) regulations in financial system, and contributed the Elliptic dataset for illicit activity detection in Bitcoin [22]. Ranshous et al. introduced the idea of motifs in directed hypergraphs and recognized some specific laundering patterns for Bitcoin exchanges [23]. Bitconeview, a visualization tool for Bitcoin, was proposed to visualize how and when an address mixes its money [24]. 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 [7]. However, this work is in lack of generalization for different mixing services and it only utilizes the topology information of transaction network. Inspired by a related study about detecting Ponzi schemes on Ethereum [25], in this paper, we propose features from multi-level, trying to discover the transaction patterns of mixing services for 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 [9], 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 [26] and functional properties. Since many interactions between objects are intermittent rather than persistent [27], network motifs combined with temporal information were proposed to characterize dynamic homogeneous network [10], and also had an extensive version in heterogeneous information network [28]. Recently, there are many studies utilized network motifs in blockchain transaction network mining tasks, such as price prediction [29], [30], network property analysis [31], exchange pattern mining [23] and so on. Network attributes play important roles in network mining tasks [11], nevertheless, few work have been conducted to considering in the construction of specific attributes.

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 Address-Address Interaction Network (AAIN) and a heterogeneous Transaction-Address Interaction Network (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 "dirty" 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 specific time window. Since the available data is 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 conduct a more thorough study by considering more complex mixing strategies.

REFERENCES

[1] S. Nakamoto. 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 Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 9, pp. 1421–1428, 2018.

[3] W. Chen and Z. Zheng, "Blockchain data analysis: A review of status trends and challenges," *Journal of Computer Research and Development*, vol. 55, no. 4, pp. 1853–1870, 2018.

[4] B. E. Howell and P. H. Potgieter. Industry self-regulation of cryptocurrency exchanges. [Online]. Available: http://dlc.dlib.indiana.edu/dlc/bitstream/handle/10535/10528/20190613_CryptoExchanges.pdf

[5] M. Mser, R. Bhme, and D. Breuker, "An inquiry into money laundering tools in the Bitcoin ecosystem," in *Proceedings of the 2013 Anti-Phishing Working Group eCrime Researchers Summit*. San Francisco: IEEE, 2013, pp. 1–14.

[6] Y. Fanusie and T. Robinson. 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

[7] M. A. Prado-Romero, C. Doerr, and A. Gago-Alonso, "Discovering Bitcoin mixing using anomaly detection," in *Proceedings of the 22nd Iberoamerican Congress on Pattern Recognition*. Valparaiso, Chile: Springer, 2017, pp. 534–541.

[8] B. Liu, Y. Dai, X. Li, W. S. Lee, and S. Y. Philip, "Building text classifiers using positive and unlabeled examples," in *Proceedings of the 3rd IEEE International Conference on Data Mining*, vol. 3. Melbourne, Florida, USA: Citeseer, 2003, pp. 179–188.

[9] 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.

[10] A. Paranjape, A. R. Benson, and J. Leskovec, "Motifs in temporal networks," in *Proceedings of the 10th ACM International Conference on Web Search and Data Mining*. Cambridge, UK: ACM, 2017, pp. 601–610.

[11] 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. 47 707–47 716, 2018.

[12] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proceedings of the 8th IEEE International Conference on Data Mining*. Pisa, Italy: IEEE, 2008, pp. 413–422.

[13] M. Kubat, S. Matwin *et al.*, "Addressing the curse of imbalanced training sets: one-sided selection," in *Proceedings of the 14th International Conference on Machine Learning*, vol. 97. Nashville, Tennessee, USA: Nashville, USA, 1997, pp. 179–186.

[14] Y. Tang, Y.-Q. Zhang, N. V. Chawla, and S. Krasser, "SVMs modeling for highly imbalanced classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 1, pp. 281–288, 2008.

[15] B. Tang, H. He, P. M. Baggenstoss, and S. Kay, "A Bayesian classification approach using class-specific features for text categorization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 6, pp. 1602–1606, 2016.

[16] M. Vasek and T. Moore, "There's no free lunch, even using Bitcoin: Tracking the popularity and profits of virtual currency scams," in *Proceedings of the 19th International Conference on Financial Cryptography and Data Security*. Puerto Rico: Springer, 2015, pp. 44–61.

[17] P. Tasca, A. Hayes, and S. Liu, "The evolution of the bitcoin economy: Extracting and analyzing the network of payment relationships," *The Journal of Risk Finance*, vol. 19, no. 2, pp. 94–126, 2018.

[18] N. Christin, "Traveling the silk road: A measurement analysis of a large anonymous online marketplace," in *Proceedings of the 22nd International Conference on World Wide Web*. Rio de Janeiro, Brazil: ACM, 2013, pp. 213–224.

[19] T. Chen, Y. Zhu, Z. Li, J. Chen, X. Li, X. Luo, X. Lin, and X. Zhange, "Understanding Ethereum via graph analysis," in *Proceedings of IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. Honolulu, HI, USA: IEEE, 2018, pp. 1484–1492.

[20] D. S. H. Tam, W. C. Lau, B. Hu, Q. Ying, D. M. Chiu, and H. Liu, "Identifying illicit accounts in large scale e-payment networks - A graph representation learning approach," *CoRR*, vol. abs/1906.05546, 2019. [Online]. Available: http://arxiv.org/abs/1906.05546

[21] W. Chen, J. Wu, Z. Zheng, C. Chen, and Y. Zhou, "Market manipulation of Bitcoin: Evidence from mining the mt. gox transaction network," in *Proceedings of IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. Paris, France: IEEE, 2019, pp. 964–972.

[22] M. Weber, G. Domeniconi, J. Chen, D. K. I. Weidele, C. Bellei, T. Robinson, and C. E. Leiserson, "Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics," *CoRR*, vol. abs/1908.02591, 2019. [Online]. Available: http://arxiv.org/abs/1908.02591

[23] S. Ranshous, C. A. Joslyn, S. Kreyling, K. Nowak, N. F. Samatova, C. L. West, and S. Winters, "Exchange pattern mining in the Bitcoin transaction directed hypergraph," in *Proceedings of the 21st International Conference on Financial Cryptography and Data Security*. Malta: Springer, 2017, pp. 248–263.

[24] 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 *Proceedings of the 2015 IEEE Symposium on Visualization for Cyber Security*, Chicago, Illinois, USA, pp. 1–8.

[25] W. Chen, Z. Zheng, J. Cui, E. Ngai, P. Zheng, and Y. Zhou, "Detecting ponzi schemes on Ethereum: Towards healthier blockchain technology," in *Proceedings of the 2018 World Wide Web Conference*. Lyon, France: International World Wide Web Conferences Steering Committee, 2018, pp. 1409–1418.

[26] A. R. Benson, D. F. Gleich, and J. Leskovec, "Higher-order organization of complex networks," *Science*, vol. 353, no. 6295, p. 163, 2016.

[27] P. Holme, "Network reachability of real-world contact sequences," *Physical Review E*, vol. 71, no. 2, p. 046119, 2005.

[28] Y. Li, Z. Lou, Y. Shi, and J. Han, "Temporal motifs in heterogeneous information networks," in *Proceedings of the 14th International Workshop on Mining and Learning with Graphs*. London, UK: ACM, 2018.

[29] C. G. Akcora, A. K. Dey, Y. R. Gel, and M. Kantarcioglu, "Forecasting bitcoin price with graph chainlets," in *Proceedings of the 22nd Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Melbourne: Springer, 2018, pp. 765–776.

[30] N. C. Abay, C. G. Akcora, Y. R. Gel, U. D. Islambekov, M. Kantarcioglu, Y. Tian, and B. M. Thuraisingham, "Chainnet: Learning on blockchain graphs with topological features," *CoRR*, vol. abs/1908.06971, 2019. [Online]. Available: http://arxiv.org/abs/1908.06971

[31] P. Moreno-Sanchez, N. Modi, R. Songhela, A. Kate, and S. Fahmy, "Mind your credit: Assessing the health of the ripple credit network," in *Proceedings of the 2018 World Wide Web Conference*. Lyon, France: International World Wide Web Conferences Steering Committee, 2018, pp. 329–338.