An Evaluation of Bitcoin Address Classification based on Transaction History Summarization

Yu-Jing Lin*, Po-Wei Wu*, Cheng-Han Hsu*, I-Ping Tu†, and Shih-wei Liao*
* *Department of Computer Science, National Taiwan University, Taiwan*
† *Institute of Statistical Science, Academia Sinica, Taiwan*
r06922068@ntu.edu.tw

***Abstract*—Bitcoin is a cryptocurrency that features a distributed, decentralized and trustworthy mechanism, which has made Bitcoin a popular global transaction platform. The transaction efficiency among nations and the privacy benefiting from address anonymity of the Bitcoin network have attracted many activities such as payments, investments, gambling, and even money laundering in the past decade. Unfortunately, some criminal behaviors which took advantage of this platform were not identified. This has discouraged many governments to support cryptocurrency. Thus, the capability to identify criminal addresses becomes an important issue in the cryptocurrency network. In this paper, we propose new features in addition to those commonly used in the literature to build a classification model for detecting abnormality of Bitcoin network addresses. These features include various high orders of moments of transaction time (represented by block height) which summarizes the transaction history in an efficient way. The extracted features are trained by supervised machine learning methods on a labeling category data set. The experimental evaluation shows that these features have improved the performance of Bitcoin address classification significantly. We evaluate the results under eight classifiers and achieve the highest Micro-F1 / Macro-F1 of 87% / 86% with LightGBM.**

***Index Terms*—bitcoin, blockchain, classification, moments, transaction history summarization**

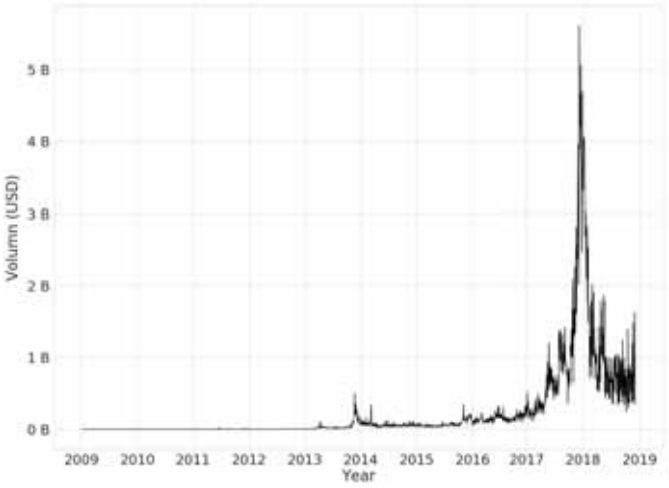


Fig. 1: Daily Transaction Volume of Bitcoin in USD. The estimated value of transactions in USD are retrieved from *Blockchain.com*[1]. Note that the change of each transaction is excluded.

I. Introduction

Since Bitcoin was released in 2008 [1], it has captivated the world with its autonomy and decentralization. Bitcoin is designed as a digital currency system based on peer-to-peer networks instead of a central administration like banks. The proof-of-work protocol allows participants to reach consensus over the distributed network. In addition, all transactions are verified by full nodes and stored in blocks which are chained together by associating previous block header hash. In addition, each block holds the Merkle root of its transactions, which is a kind of fingerprint of transactions, in order to prevent evildoers from tampering data on the blockchain. The properties of immutability, decentralization, data integrity, security of Bitcoin make itself a trustworthy digital currency.

As the pioneer of thousands of cryptocurrencies, Bitcoin is the most valuable one in terms of market capitalization (market cap). [2] reports that Bitcoin has a market cap of around 59 billion USD, dominating over half of the total market cap of all cryptocurrencies. Moreover, the transaction volume per day on the Bitcoin network in Figure 1 raised to billions of US dollars since the mid of 2017 and even once bumped up to 5 billion USD per day. The profitable potential of Bitcoin has attracted people to engage in various activities on Bitcoin, such as payment, investment, gambling, and even laundering.

Some criminal behaviors, such as laundering and frauds, are encouraged by Bitcoin's anonymity. Although Bitcoin is usually described as an anonymous currency, it is actually pseudo-anonymous [3], i.e, it is hard to link a user to an address by exploring transactions on the blockchain. However, some signs implying the link between addresses and users or between addresses and their usages can be observed. Therefore, there exist messages to possibly identify the criminal behaviors from benign ones.

In this paper, we propose using the *transaction moments* and a series of *extra statistics* as strong features of *transaction history summary* to identify abnormal addresses. We evaluate the proposed features with eight classifiers using 10-fold cross-validation. We only use the relevant transactions, which is

[1]https://www.blockchain.com/charts/estimated-transaction-volume-usd?timespan=all

978-1-7281-1328-9/19/$31.00 ©2019 IEEE

defined as transaction history, of an address or an entity. It is more efficient because the size of data to traverse for a graph pattern might exponentially grow while that of transaction history is only proportional to the relevant transactions of an address or an entity. We finally achieve a Micro-F1 score of 87% and a Macro-F1 of 86% with LightGBM [4] in the address-based scheme, which is comparable to the results of prior works [5], [6].

This paper is organized as follows: in Section II, we explain Bitcoin network. Next, we investigate several related research on identification, or de-anonymization, of Bitcoin addresses in Section III. Section IV illustrates our proposed method: transaction history summary. Data collection, feature extraction, classification, and other training details are elaborated in Section V. We then evaluate the classification result in Section VI and conclude the paper with Section VII.

II. Bitcoin Network

In the Bitcoin network, transactions specify the number of bitcoins, which are the currency in Bitcoin, to be taken from one address and the number to be transferred to another address. Each transaction can hold multiple inputs and multiple outputs as long as the total amount of inputs is greater than or equal to that of outputs. When making a payment, a user signs the transaction with his private key so as to prove his ownership of the bitcoins to be spent.

The common unit of Bitcoin is bitcoin (BTC) while each bitcoin is divisible to the eighth decimal place. A BTC can be split into 100,000,000 units, called satoshis, which are the smallest unit of Bitcoin. Most transactions contain transaction fees, which will be transferred to the miner's address as rewards for their proof-of-works. Although transaction fee is not obliged, transactions without any transaction fee or with a lower fee than usual are less likely to be packed to a block by miners. The identities in Bitcoin are private keys. Each private key generates a public key and an address used to public identification. Anyone with a private key is able to spend all bitcoins corresponding to its address. On the other hand, a user can hold an arbitrary number of private keys so it is hard to link an address to a person.

Figure 2 shows an example of a transaction that specifies a payment with payback from Alice to Bob. Alice wants to send 2.5 BTC to Bob. She first gathers two of her UTXOs (unspent transaction outputs) holding 2 BTC and 1 BTC, which are somehow received from other addresses. Then Alice signs the two inputs and specifies the outputs as 2.5 BTC to Bob, 0.005 BTC as the transaction fee, and 0.495 BTC back to herself. After the transaction is confirmed on the blockchain, Bob is able to spend the 2.5 BTC.

The Bitcoin network can be viewed as a large composition of transactions. Each transaction is composed of one or multiple inputs and one or multiple outputs. It also records other information such as generation time of blocks. Therefore, we can analyze the whole Bitcoin networks by traversing all the blocks and extract useful information from them.

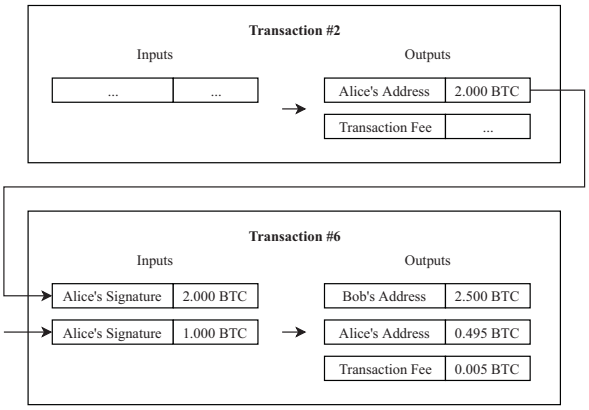


Fig. 2: An example of Bitcoin transactions.

III. Related Work

There are numerous publications that aimed at Bitcoin network analysis, entity identification, and address de-anonymization. Unlike the descendants emerging in recent years, e.g. Ethereum [7], EOS [8], etc., which support smart contract programming, Bitcoin works as a pure transaction ledger. The simplicity makes Bitcoin easy to analyze since the network only contains a bunch of transactions. While data on Bitcoin is open to everyone, some experiments additionally leverage off-chain information such as address tags, indicating potential owners or possible usages of them. These supplementary data put analyzing the Bitcoin network with supervised learning methods into practice.

An intuitive investigation on Bitcoin transactions is to study its transaction flow. [9] studied the characteristics of transaction graphs and clustered addresses which might belong to the same entity. Then it is possible to compact a transaction graph into an entity graph. On the other hand, the corresponding input and output addresses confounded by mixing services can be partially understood by transaction graph analysis [10].

Several address clustering heuristics [11], [12] are proposed to link addresses to entities, which represent groups of addresses owned by the same people or the same organizations. The partial linkability between addresses and entities is revealed by several characteristics on the Bitcoin network. Although there is currently no way to link an arbitrary address to its user in the real world, the associated entities are able to be evaluated with off-chain information such as tags (mining pool, exchange wallet, etc.).

A number of studies focused on clustering addresses by unsupervised learning methods. [13]–[15] extract features and cluster address based on statistics of address and patterns of transaction flow in order to detect fraudulent activity in Bitcoin transactions. In these works, k-means [16] and its variants are adopted to classify features extracted from Bitcoin addresses.

Other studies solve the entity identification problem by supervised learning methods. [17] classifies cybercriminal entities by supervised learning methods on collected labeled Bitcoin addresses. [18] train classifiers to detect Ponzi schemes

in Bitcoin. To deal with imbalanced data, sampling-based approach and cost-sensitive approach are considered simultaneously in [18]. To reduce anonymity of Bitcoin by predicting yet-unidentified addresses, [19] trained classifiers with synthetic minority over-sampling technique [20] on imbalanced data.

[21] introduces the idea of *motifs* in directed hypergraphs, defining exchange patterns of addresses. In [6], the graph-based features *motifs* are then combined with address features, entity features, temporal features, and centrality features to identify Bitcoin entity categories.

Besides graph patterns, transaction history can also provide information for address identification. In [22], a set of features are proposed to summarize the transaction history and to identify addresses associated with HYIP based on supervised learning algorithms. These features are extended to identify seven types of Bitcoin-enabled services [5].

IV. Proposed Method

Our work stands on [5], which aimed to identify Bitcoin-enabled service categories based on transaction history summary. We notice the effectiveness of transaction history summary and proposed a series of features to elevate it in various aspects.

A *transaction history summary* is derived from the transaction history, considering merely the direct relevant transactions of a given address or entity. Since the related transactions to an address or an entity contain many redundant fields which cause overhead on the classification model, we extract features from the transactions as transaction history summary. We wish to demonstrate the effectiveness of different types of features. Accordingly, we propose the following three feature types:

- Basic Statistics referred from [5] as the baseline features for address classification,
- Extra Statistics replenishing the address statistics,
- Moments corresponding to transaction distributions.

The following subsections provide more details on each type of features.

A. Basic Statistics

The originally proposed features [5] are composed of eight statistical characteristics. Basic statistics include the number of transaction per day, the ratio of received, coinbase and payback transactions to all transactions, the frequencies of different orders of magnitude of transferred bitcoins in spent transactions and in received transactions, and the average numbers of inputs and outputs in the spent transactions. These features are counted numbers divided by duration or a total number. As a result, they characterize transaction history in the aspect of frequency. In [5], accuracies of 70% and 72% are achieved by these basic statistical features using a random forest classifier in the address-based scheme and the entity-based scheme.

TABLE I
Moments.

Name	Meaning
1^{st} *moment*	measure of location
2^{nd} *moment*	measure of spread
3^{rd} *moment*	measure of symmetry
4^{th} *moment*	measure of peakedness

B. Extra Statistics

Yet there are some characteristics not captured by the basic statistics. We complement the features with extra statistics. The active duration of a series of transactions is defined as $lifetime$, which is the difference between the date of the earliest transaction and that of the latest transaction in terms of the number of days. Mixers tend to have short $lifetimes$ because they are usually disposed of after use. The total received bitcoins and spent bitcoins are taken into consideration, denoted as BTC_{received} and BTC_{spent}. Likewise, the total received and spent money in US dollars are involved as USD_{received} and USD_{spent}. The original values and equivalent values in the real world are both considered in this way. Note that here we count US dollars by converting the amount in each transaction according to its rate at that time.

In addition to the active duration and total money statistics, the numbers of all types of transactions are included as n_{TX}, n_{spent}, n_{received}, n_{coinbase}, and n_{payback}. Furthermore, the balances after each transaction are also helpful. We calculate the mean and standard deviation of the balances in BTC and USD to be the last four features, which are $\mu_{\text{balance\_btc}}$, $\sigma_{\text{balance\_btc}}$, $\mu_{\text{balance\_usd}}$, and $\sigma_{\text{balance\_usd}}$. The reason is that, for example, a mixer might send a large number of bitcoins after it receives them, so its balances would have a large standard deviation than addresses of other categories have.

C. Transaction Moments

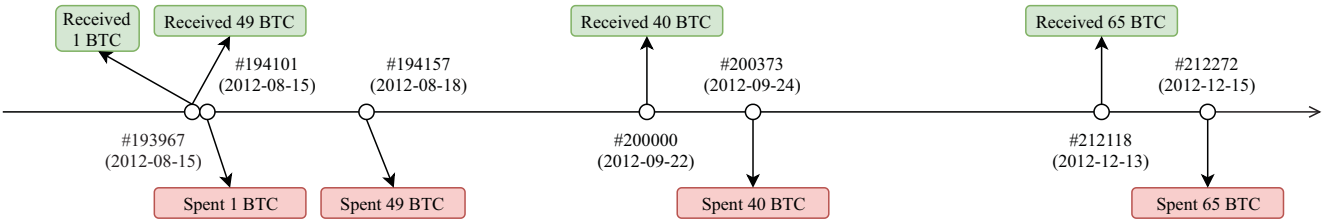
Addresses of different categories have different time distributions of transactions. However, there are no temporal features in the aforementioned statistical features. In order to capture the temporal information, we propose using *Transaction Moments* to encode temporal information as features.

A moment is a quantitative measure of a distribution function. Table I illustrates the distribution behaviors measured by moments of different orders. Generally, four orders are commonly used to describe the shape of a distribution. Therefore, we define first moment (mean), second central moment (variance), third standardized moment (skewness), and fourth standardized moment (kurtosis) of transaction distributions as *Transaction Moments* (m_n).

Definition 1. Moment. The n-th moment μ_n is defined on a real-valued continuous function f as

$$\mu_n = \int_{-\infty}^{\infty} (x - c)^n f(x) dx$$

, where c is a central constant. c is usually zero while central moment uses c as the mean of x.



Transactions happen in block heights of 193967, 193967, 194101, 194157, 200000, 200373, 212118, 212272.

$$m_1 = \mathrm{E}[X] = 193967 * \frac{2}{8} + 194101 * \frac{1}{8} + 194157 * \frac{1}{8} + 200000 * \frac{1}{8} + \cdots + 212272 * \frac{1}{8} = 200119.375$$

$$m_2 = \mathrm{E}[(X-\mu)^2] = (193967 - 200119.375)^2 * \frac{2}{8} + \cdots + (212272 - 200119.375)^2 * \frac{1}{8} \cong 54900757$$

$$m_3 = \mathrm{E}\left[(\frac{X-\mu}{\sigma})^3\right] \cong \left(\frac{193967 - 200119.375}{7409.5}\right)^3 * \frac{2}{8} + \cdots + \left(\frac{212272 - 200119.375}{7409.5}\right)^3 * \frac{1}{8} \cong 0.807084$$

$$m_4 = \mathrm{E}\left[(\frac{X-\mu}{\sigma})^4\right] \cong \left(\frac{193967 - 200119.375}{7409.5}\right)^4 * \frac{2}{8} + \cdots + \left(\frac{212272 - 200119.375}{7409.5}\right)^4 * \frac{1}{8} \cong 1.989782$$

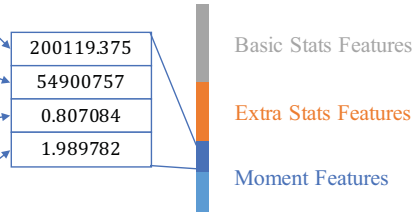


Fig. 3: An example of transaction history. The upper timeline demonstrates the transaction history of address *15L23mj1TnFa9trXdpQ83iXrGzVd1byKUG*. It is available on public blockchain explorer such as *Blockchain.com*[2]and *Blockcypher.com*[3]. The history contains transactions recorded in the 193967th, 194101st, 194157th, 200000th, 200373rd, 212118th, 212272nd blocks. The overall transaction moments are calculated based on the listed transaction history.

Definition 2. Moment of a continuous random variable. If f is a probability density function and F is its cumulative probability distribution function, the n-th moment is

$$\mu_n = \mathrm{E}[X^n] = \int_{-\infty}^{\infty} x^n dF(x)$$

, where X is a continuous random variable with probability density function f.

Definition 3. Expected value of a discrete random variable. Let X be a discrete random variable with support R_X and probability mass function p_X. The expected value of X is

$$\mathrm{E}[X] = \sum_{x \in R_X} x p_X(x)$$

Definition 4. Moment of a discrete random variable. If p_X is a probability mass function of a discrete random variable X, the n-th moment is derived from definition 2 and 3 as

$$\mu_n = \mathrm{E}[X^n] = \sum_{x \in R_X} x^n p_X(x)$$

Given the above definitions, we consequently characterize the occurrence time of transactions of an address or an entity as a discrete random variable, which has a probability mass function. Note that the occurrence time here is actually the height of block the transaction subsumed. The mathematical forms of the Transaction Moments we adopt are defined as follows.

[2]https://www.blockchain.com/explorer
[3]https://live.blockcypher.com

1) First Moment: The first moment is identical with mean.

$$m_1 = \mathrm{E}[X] \tag{1}$$

2) Second Central Moment: We take the second "central" moment as our second moment feature, which is also known as variance.

$$m_2 = \mathrm{E}[(X-\mu)^2], \tag{2}$$

where μ is the expected value of X, i.e., $\mu = m_1$.

3) Third Standardized Moment: The third moment is centralized and standardized as the term says.

$$m_3 = \mathrm{E}[(\frac{X-\mu}{\sigma})^3], \tag{3}$$

where μ is the expected value of X and σ is the standard deviation of X, i.e., $\mu = m_1$ and $\sigma = \sqrt{m_2}$.

4) Fourth Standardized Moment: The fourth moment is also centralized and standardized.

$$m_4 = \mathrm{E}[(\frac{X-\mu}{\sigma})^4], \tag{4}$$

where μ is the expected value of X and σ is the standard deviation of X, i.e., $\mu = m_1$ and $\sigma = \sqrt{m_2}$.

Figure 3 depicts an example of transaction history and demonstrates how to calculate the overall transaction moments of a given address. In this paper, we measure six transaction moments in total. They are moments of *overall transactions*,

TABLE II
Dataset Details.

Category	# of Entities	# of Addresses	# of TXs
Exchange	158	10,466	5,701,261
Faucet	61	340	181,602
Gambling	90	6,733	6,536,088
HYIP	956	2,026	377,084
Market	18	1,900	93,930
Mixer	32	3,199	49,064
Pool	38	1,644	274,168
Total	1,353	26,308	13,084,546

coinbase transactions, *spent transactions*, *reveived transactions*, *payback transactions* as well as *transaction intervals* which are the intervals between transactions in chronological order in terms of block heights. The moments are then concatenated with basic statistics and extra statistics to serve as the transaction history summary.

V. Experiments

To evaluate how effective the proposed features are, we design an experiment of Bitcoin category classification based on addresses and entities. Firstly, we collected labeled data of address-label pairs and fetched all transactions associated with the addresses. The addresses and entities are then be summarized into features with the use of these data. We trained eight supervised classifiers on the extracted features and evaluate the results by average Micro-F1 scores and average Macro-F1 scores of 10-fold cross-validation.

A. Collect Data

We leverage the dataset collected by [5] in order to facilitate the comparison between our method and the previous work. As described in Table II, the dataset contains totally 26,313 addresses with labels and owners, which are derived from a simple heuristic, naming *multi-input transactions* [11], *shared-send clustering* [12], *co-spend clustering* [19], or *common spending* [6]. The idea is that the addresses of inputs in a transaction belong to the same entity because spending bitcoins needs the signature of the owner's private key.

There are 7 categories in total while the data are imbalanced in both address-based scheme and entity-based scheme. We collected the relevant transactions from 2009-01-03 to 2018-06-30 of the addresses, which are over 13 million transactions in total. Some invalid or undecodable transactions are filtered out and addresses/entities containing zero valid transactions are also removed in advance.

B. Summarize Transaction Histories

We summarize all transaction histories by address and by entity respectively. The selected features are listed in Table III, which is divided into three parts: basic statistical features, extra statistical features, and moment features. Their dimensions are 26, 14, and 24 respectively while summed up to be 64 dimensions.

The three types of coinbase, spent, received are mutually exclusive. Specifically, a transaction is assigned as a coinbase transaction if it contains a coinbase input. If a transaction has no coinbase input and the address appears in some of its inputs, the transaction is identified as a spent transaction. Otherwise, if the address appears only in the outputs, it is assigned as a received transaction.

With regard to the moment features we propose, we measure the moments of overall transaction distribution, spent transaction distribution, received transaction distribution, coinbase transaction distribution, payback transaction distribution, and transaction interval distribution. We deem that how often an address or an entity has transactions is important in order to reveal what category it is. In addition, not only the frequency, moments of transaction distributions and transaction interval distributions can characterize behaviors such as an address

TABLE III
The List of Summarized Features from Transaction History.

Feature	Description
f_{TX}	The frequency of transactions, defined as number of all transactions per day in the address/entity's lifetime.
r_{received}	The ratio of received transactions to all transactions.
r_{coinbase}	The ratio of coinbase transactions to all transactions.
$f_{\mathrm{spent}}(10^i)$	The frequency of digit i in USD appeared in spent transactions, where $i \in (10^{-3}, 10^{-2}, \ldots, 10^{6})$.
$f_{\mathrm{received}}(10^i)$	The frequency of digit i in USD appeared in received transactions, where $i \in (10^{-3}, 10^{-2}, \ldots, 10^{6})$.
r_{payback}	Payback ratio defined as the ratio of Bitcoin addresses that appear in both inputs and outputs.
$\bar{N}_{\mathrm{inputs}}$	The mean value of the number of inputs in the spent transactions.
$\bar{N}_{\mathrm{outputs}}$	The mean value of the number of outputs in the spent transactions.
	Basic Statistics
$lifetime$	The duration between the first transaction and the last transaction in terms of days.
BTC_{spent}	Total spent BTC.
BTC_{received}	Total received BTC.
USD_{spent}	Total spent USD, which are converted based on daily BTC/USD rates from *Coinmarketcap.com* [2].
USD_{received}	Total received USD, which are converted based on daily BTC/USD rates from *Coinmarketcap.com* [2].
n_{TX}	The number of transactions.
n_{spent}	The number of spent transactions.
n_{received}	The number of received transactions.
n_{coinbase}	The number of coinbase transactions.
n_{payback}	The number of payback transactions.
$\mu_{\mathrm{balance\_btc}}$	The mean value of balance in BTC after each transaction.
$\sigma_{\mathrm{balance\_btc}}$	The standard deviation of balance in BTC after each transaction.
$\mu_{\mathrm{balance\_usd}}$	The mean value of balance in USD after each transaction.
$\sigma_{\mathrm{balance\_usd}}$	The standard deviation of balance in USD after each transaction.
	Extra Statistics
$m_{\mathrm{n,overall}}$	The moments of overall transaction distribution.
$m_{\mathrm{n,spent}}$	The moments of spent transaction distribution.
$m_{\mathrm{n,received}}$	The moments of received transaction distribution.
$m_{\mathrm{n,coinbase}}$	The moments of coinbase transaction distribution.
$m_{\mathrm{n,payback}}$	The moments of payback transaction distribution.
$m_{\mathrm{n,interval}}$	The moments of transaction interval distribution.
	Moments

active in the beginning but listless in recent days, and even periodic behaviors. Whether a transaction relates to spending Bitcoins, receiving Bitcoins, or even spending Bitcoins back to the spender matters. They are also indications of the usages of their addresses.

In [5], only up to 1000 successive transactions of an entity are summarized due to the huge data size. We instead collected all related transactions of the addresses in the dataset so as to extract features from all transactions of an address or an entity. The whole history is characterized into the extracted transaction history summary.

C. Train Classifiers

The extracted features, or transaction history summaries of an address, are then classified by machine-learning-based algorithms. We would like to compare our features to recent studies in Bitcoin address classification. Most of them classified the extracted features with several machine learning methods. Therefore, in this work, we evaluate them with eight classifiers: Logistic Regression, Perceptron [23], Support Vector Machine (SVM) [24], Adaptive Boosting with Decision Tree (AdaBoost) [25]–[27], Random Forest (RF) [28], Extreme Gradient Boosting (XGBoost) [29], Light Gradient Boosting Machine (LightGBM) [4], and Neural Network [30].

D. Implementation Details

For the first seven classifiers, we exploit the Python machine learning library - Scikit-learn [31]. Each classifier is tuned by grid search with 10-fold cross-validation in order to find a good set of parameters respectively. The decision tree-based methods are not affected by data normalization. We simply normalize the features with division by the maximum absolute value of each dimension for the classifiers that are not based on decision trees (Logistic Regression, Perceptron, and SVM).

The neural networks are implemented with Keras [32]. The architecture is composed of four fully-connected layers of hidden size 512 with batch normalization and dropout regularization. The model is followed by an output fully-connected layer of size 7 at the end. Similarly, we normalize the data in advance since neural networks are sensitive to input data.

As can be seen in Table II, the category imbalance exists in both schemes. To deal with the imbalance, we adopt the stratified random sampling, split training set and validating set, and take up the cost-sensitive learning to train classifiers. The weight of each sample is calculated by

$$w_i = k/p_i \tag{5}$$

, where k is a constant usually defined as $\frac{1}{\#\ of\ categories}$ and p_i is the probability of the category of the sample.

Another pitfall is the bias in transaction temporal distribution. We observe a bias among different categories caused by data collection. In Figure 4, it is obvious that some categories like Gambling, Faucet, and Pool are distinguishable merely by their first moments of the distributions. Therefore, we

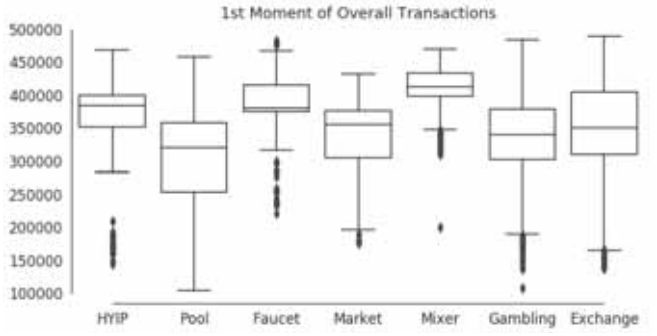


Fig. 4: The first moment of overall transactions of addresses from seven categories. The upper, middle, lower lines on the boxes represent the quartiles and the highest datum and lowest datum are 1.5 IQR from the Q1 and from the Q3. The data points out of whiskers are plotted as dots.

normalize the distributions of overall transactions, coinbase transactions, spent transactions, received transactions, and payback transactions, while excluding transaction intervals. In practice, we subtract random variables by their minimum value as follows.

$$X' = X - min(X), \tag{6}$$

where $min(X)$ is the minimum value in X.

The first moment of Equation 1 is replaced by

$$m_1' = \mathrm{E}[X'] = \mathrm{E}[X] - min(X) \tag{7}$$

, which we call the first min-shifted moment of X.

Other moments are **central** moments so they are not affected by the min-shift. For the example in Figure 3, the first min-shifted moment is computed as $m_1 = 200119.375 - 193967 = 6152.375$.

VI. Evaluation and Discussion

To show the effectiveness of the proposed methods, we evaluate on (i) Micro-F1 scores and Macro-F1 scores, (ii) selected features, (iii) the confusion matrix, and (iv) important features. For (i), we compare the results among eight supervised machine learning classification algorithms on the transaction history summaries of the labeled addresses and entities. Then the same metrics (accuracy, precision, and F1-score) are used to evaluate the features (ii) that we select by an ablation study on different feature combinations.

The best Micro-F1 scores are 91% for the entity-based scheme and 87% for the address-based scheme. However, the Macro-F1 scores are 78% and 86% instead. Table IV shows the detailed results of two schemes with Micro-F1 scores and Macro-F1 scores. A Micro-F1 computes the overall average F1 score over the testing set, whereas a Macro-F1 computes the F1 score independently for each category and take the average, treating all categories equally. We suspect that the large difference between entity-based F1 scores suffers

TABLE IV
Results of Supervised Classifiers with Full Features.

Method	Entity-based Scheme		Address-based Scheme	
	Micro-F1	Macro-F1	Micro-F1	Macro-F1
Logistic Regression	0.73	0.60	0.48	0.45
Perceptron	0.69	0.53	0.36	0.35
SVM	0.56	0.46	0.47	0.46
AdaBoost	0.30	0.30	0.36	0.36
Random Forest	0.90	0.73	0.83	0.81
XGBoost	0.90	0.77	0.83	0.82
LightGBM	0.90	0.75	**0.87**	**0.86**
Neural Network	**0.91**	**0.78**	0.83	0.81

from the data imbalance and data scarcity. For the entity-based scheme, Market has only 1 or 2 samples in the testing when evaluating by 10-fold cross-validation. Consequently, we focus on the address-based scheme in further experimental evaluations.

The Random Forest, XGBoost, LightGBM, and Neural Network are the best four machine learning methods among the eight classifiers in our experiment. Comparing to the prior work [5], that has achieved an accuracy of 72% in the entity-based scheme and 70% in the address-based scheme, we achieve a better result by Random Forest with the proposed features. In general, we showed that the categories are identified more accurately by classifiers working with our proposed features.

Although the best result in entity-based is achieved by neural networks, LightGBM performs most stably in both schemes. As a result, we use the best model of LightGBM to illustrate (iii) confusion matrix and (iv) important features in the following part. On the other hand, with moments and extra statistics, as seen in the table, the decision tree-based classifier and perceptron work better on address-scheme while neural network, SVM and logistic regression have better results on the entity-based scheme.

The performance of LightGBM with different combinations of features is presented in Table V. The features are divided into three types: Basic Statistics, Extra Statistics and Moments. Our result shows the effectiveness of feature combinations. Evaluated with any combinations of features, the result is better than evaluated with any single feature. It also implies that the basic statistics and extra statistics capture two main behaviors, whereas moments remedy the transaction history summary. Although moments alone do not work as well as statistical features, moments boost up the F1 scores when combined with other features. We achieve the best result with basic statistics, extra statistics, and moments mixed together.

TABLE V
Ablation Study of Basic Statistics, Extra Statistics and Moments in Address-based Scheme.

Method			Results	
B	E	M	Micro-F1	Macro-F1
√			0.79	0.78
	√		0.77	0.74
		√	0.66	0.57
√	√		0.86	0.85
√		√	0.84	0.82
√	√	√	**0.87**	**0.86**

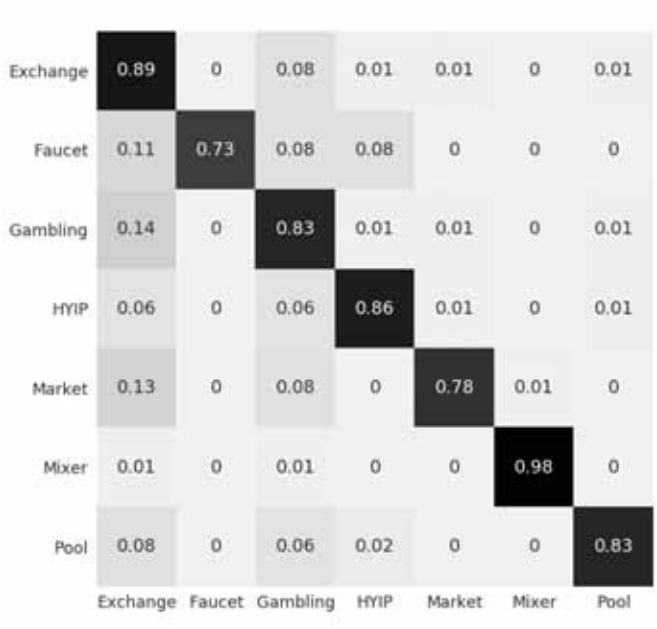


Fig. 5: The confusion matrix of LightGBM trained with all features in the address-based scheme. The values in grids are categorical accuracy, where each row is supposed to be summed up as 1. Besides, the darkness of each grid is proportional to its value.

The confusion matrix of LightGBM with all features in the address-based scheme is depicted in Figure 5. Most categories are classified well, indicating that the classifier works well on these categories. The classification accuracy of Mixer is even 96% and the second high one is 84% of Exchange. Other categories achieve at least 70% in terms of accurate classification.

Table VI illustrates the 20 most important features in LightGBM classifier, which achieves best result in the address-based scheme. The features are sorted by the information gain importance [33] from largest to smallest. As can be seen from the table, six out of the ten features are proposed by us. Although the moment features are less important than the other two kinds, they take one-fourth of the top twenty features. The

TABLE VI
Top 20 Important Features According to the Model.

(a) The top 10 features.

Feature Name	Feature Type
f_{TX}	Basic Stats
$\bar{N}_{\text{outputs}}$	Basic Stats
$\bar{N}_{\text{inputs}}$	Basic Stats
n_{received}	Extra Stats
$m_{1,\text{interval}}$	Moments
$\sigma_{\text{balance\_btc}}$	Extra Stats
lifetime	Extra Stats
r_{payback}	Basic Stats
$m_{1,\text{received}}$	Moments
$\mu_{\text{balance\_btc}}$	Extra Stats

(b) The 11th to 20th features.

Feature Name	Feature Type
$f_{received}$	Basic Stats
n_{spent}	Extra Stats
n_{TX}	Extra Stats
$m_{2,\text{received}}$	Moments
BTC_{spent}	Extra Stats
$\mu_{\text{balance\_usd}}$	Extra Stats
$m_{1,\text{total}}$	Moments
$m_{2,\text{total}}$	Moments
BTC_{received}	Extra Stats
$f_{\text{received}}(10^2)$	Basic Stats

extra statistics, on the other hand, appear almost the same times as the basic statistics. For more details about important features, please refer to Appendic A.

VII. Conclusion

In this work, we introduce new features as transaction history summary for Bitcoin address and entity classification. The transaction history summary is composed of basic statistics, extra statistics, and transaction moments. The basic statistics are based on the previous work [5] and capture the features in the aspect of frequency. The extra statistics additionally contain total amounts and statistical measures of transactions. The transaction moments characterize the temporal distribution of transactions as well as transaction intervals.

Our experiment showcases the performance benefits from using our proposed features for Bitcoin address/entity classification. The combinations of features make huge progress in terms of classification accuracy. Moreover, our proposed features dominate the ten most important features according to a well-trained LightGBM classifier. As the best result we achieve, the Micro / Macro F1 scores are 87% / 86% in the address-based scheme. The high accuracy in each category is indicated from measuring the similarity between Micro-F1 and Macro-F1. Also, the confusion matrix of our best result further proves it. The entity-based classification, however, suffers from data imbalance and data scarcity. Therefore, we plan to do the experiment on a larger dataset [6] in the future work so as to evaluate the entity-based scheme.

Appendix A
Feature Importance

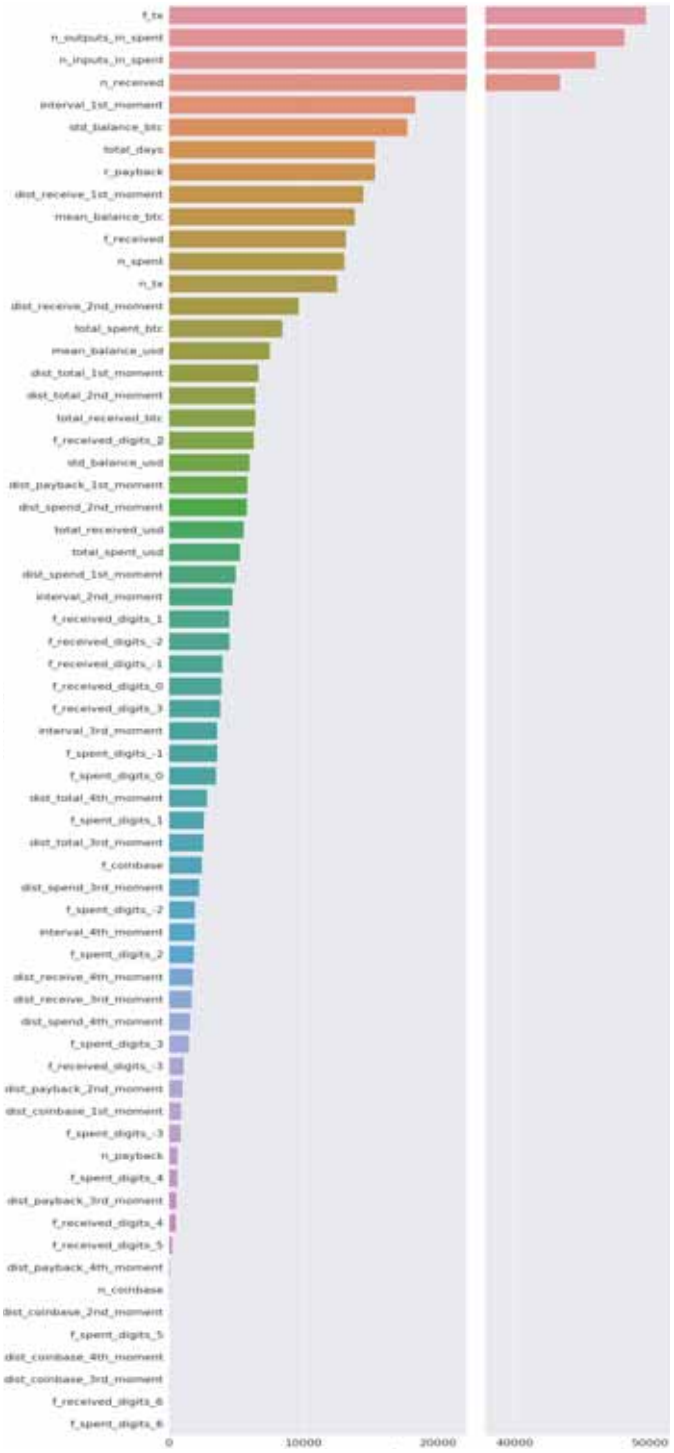


Fig. 6: Feature importance scores are reported as the total information gains of splits for each feature in LightGBM.

REFERENCES

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[2] "Cryptocurrency market capitalizations."

[3] M. Conti, S. Kumar, C. Lal, and S. Ruj, "A survey on security and privacy issues of bitcoin," *IEEE Communications Surveys & Tutorials*, 2018.

[4] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," in *Advances in Neural Information Processing Systems*, pp. 3146–3154, 2017.

[5] K. Toyoda, T. Ohtsuki, and P. T. Mathiopoulos, "Multi-class bitcoin-enabled service identification based on transaction history summarization," in *International Conference on Blockchain*, IEEE, 2018.

[6] M. Jourdan, S. Blandin, L. Wynter, and P. Deshpande, "Characterizing entities in the bitcoin blockchain," *arXiv preprint arXiv:1810.11956*, 2018.

[7] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, pp. 1–32, 2014.

[8] I. Grigg, "Eos, an introduction," *Whitepaper) " https://eos.io/documents/EOS_An_Introduction.pdf*, 2017.

[9] D. Ron and A. Shamir, "Quantitative analysis of the full bitcoin transaction graph," in *International Conference on Financial Cryptography and Data Security*, pp. 6–24, Springer, 2013.

[10] M. Moser, "Anonymity of bitcoin transactions," 2013.

[11] E. Androulaki, G. O. Karame, M. Roeschlin, T. Scherer, and S. Capkun, "Evaluating user privacy in bitcoin," in *International Conference on Financial Cryptography and Data Security*, pp. 34–51, Springer, 2013.

[12] S. Meiklejohn, M. Pomarole, G. Jordan, K. Levchenko, D. McCoy, G. M. Voelker, and S. Savage, "A fistful of bitcoins: characterizing payments among men with no names," in *Proceedings of the 2013 conference on Internet measurement conference*, pp. 127–140, ACM, 2013.

[13] D. Zambre and A. Shah, "Analysis of bitcoin network dataset for fraud," *Unpublished Report*, 2013.

[14] P. Monamo, V. Marivate, and B. Twala, "Unsupervised learning for robust bitcoin fraud detection," in *Information Security for South Africa (ISSA), 2016*, pp. 129–134, IEEE, 2016.

[15] V. R. Patil, A. P. Nikam, J. S. Pawar, and M. S. Pardhi, "Bitcoin fraud detection using data mining approach," *Journal of Information Technology and Sciences*, vol. 4, no. 2, 2018.

[16] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, pp. 281–297, Oakland, CA, USA, 1967.

[17] H. S. Yin and R. Vatrapu, "A first estimation of the proportion of cybercriminal entities in the bitcoin ecosystem using supervised machine learning," in *Big Data (Big Data), 2017 IEEE International Conference on*, pp. 3690–3699, IEEE, 2017.

[18] M. Bartoletti, B. Pes, and S. Serusi, "Data mining for detecting bitcoin ponzi schemes," *arXiv preprint arXiv:1803.00646*, 2018.

[19] M. A. Harlev, H. Sun Yin, K. C. Langenheldt, R. Mukkamala, and R. Vatrapu, "Breaking bad: De-anonymising entity types on the bitcoin blockchain using supervised machine learning," in *Proceedings of the 51st Hawaii International Conference on System Sciences*, 2018.

[20] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[21] 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 *International Conference on Financial Cryptography and Data Security*, pp. 248–263, Springer, 2017.

[22] K. Toyoda, T. Ohtsuki, and P. T. Mathiopoulos, "Identification of high yielding investment programs in bitcoin via transactions pattern analysis," in *GLOBECOM 2017-2017 IEEE Global Communications Conference*, pp. 1–6, IEEE, 2017.

[23] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain.," *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[24] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their applications*, vol. 13, no. 4, pp. 18–28, 1998.

[25] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of computer and system sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[26] R. E. Schapire, "A brief introduction to boosting," in *Ijcai*, vol. 99, pp. 1401–1406, 1999.

[27] T. Hastie, S. Rosset, J. Zhu, and H. Zou, "Multi-class adaboost," *Statistics and its Interface*, vol. 2, no. 3, pp. 349–360, 2009.

[28] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[29] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794, ACM, 2016.

[30] S. Haykin and N. Network, "A comprehensive foundation," *Neural networks*, vol. 2, no. 2004, p. 41, 2004.

[31] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[32] F. Chollet *et al.*, "Keras," 2015.

[33] G. Louppe, L. Wehenkel, A. Sutera, and P. Geurts, "Understanding variable importances in forests of randomized trees," in *Advances in neural information processing systems*, pp. 431–439, 2013.