Behavioral classification of Bitcoin addresses based on transaction history

Shayan Hamidi Dehshali, Behnam Bahrak
College of Computer and Electrical Engineering, University of Tehran
{shayanhamidi, bahrak}@ut.ac.ir

***Abstract*—User anonymity was one of the main motivations of the developers of Bitcoin. The developers achieved anonymity, more specifically pseudonymity, by assigning an address with a specified format and revealing no clue about its owner. Although the transaction history of an address cannot expose the actual user behind the address, it can disclose hints about the behavioral role of the user. These behavioral roles could belong to a centralized exchange, cyber-security service, darknet market, gambler, mining pool, peer-to-peer financial service, or tumbler.**

In this paper, we intend to classify a variety of Bitcoin addresses belonging to the stated wallet types. First, we gathered addresses and their types from wallet explorer websites and address clustering methods. Then, we extracted features from the existing transaction history of an address collected from a live full-node peer. Later, we inserted the bulk of data in a PostgreSQL database management system to retrieve data efficiently. We scripted queries to calculate primary features from the database. Furthermore, we added secondary features derived from primary ones. At last, we applied different models, such as decision tree, random forest, KNN, and XGBoost, to our training and test sets. Our results show that the best model for evaluative metrics was XGBoost, with a weighted average f1-score of 98.7%.

***Index Terms*—Bitcoin, Anonymity, Classification, Machine learning, Behavioral pattern extraction**

I. Introduction

Traditional banking systems are centralized. In other words, a third party, functioning as a central watcher, controls all the monetary affairs, user authentication, and user balances. As traditional banking systems are not transparent, these systems can be manipulated. Bitcoin, introduced in 2008 by Nakamoto, provides transparency, anonymity, and security that proves advantageous to the traditional banking systems [1]. Bitcoin utilizes a peer-to-peer network with a decentralized and distributed structure. In this network, transactions are made with the help of cryptographic mechanisms, for instance, digital signatures [2]. The history of the transactions is kept in a distributed and public ledger. Since the security mechanisms of Bitcoin utilize hash functions [3] and blockchain structure [4], forging transactions or repudiating them is impossible. Moreover, all of the users could validate the transactions. In this network, each user forms a transaction with his/her public and private keys. Also, they compute an address from the public key that must be attached to the transaction. Also, each individual can own multiple addresses since no central authenticator exists. Thus, we can assert the existence of anonymity. However, with the transparency of the transaction belonging to an address, the privacy of Bitcoin users is flawed.

User anonymity was one of the main motivations of the developers of Bitcoin. Even though this property is advantageous in many ways, it has challenged the security of our society. Many illegal markets, for example, gun shops, drugs, smugglers, and money laundering individuals, chose this platform to stay anonymous. Moreover, gambling and betting have become prevalent among Bitcoin users. Although matching a Bitcoin address to its actual user is complicated and challenging, it is possible to extract behavioral patterns from the transparent transactions on the blockchain and identify the role of the real user behind an address. This feature enables criminal activity detection and possible prevention.

The activities of an address should be tracked to detect its felonies. Referring to web pages or messages that an address appears on or finding its user's IP address are some of the techniques for revealing the true identity behind the address. [5], [6], and [7] utilize network traffic analysis to discover the IP address and transactions of a Bitcoin address. Classification, our focus in this paper, is a machine learning-based method that collects labeled addresses and compares them to the target addresses to forecast their behavioral role. As mentioned, transaction history is the only activity recorded for an address. Therefore, transaction-extracted features are the sole criteria for the classification process.

The behavioral roles that we address in this paper include central exchanges, darknet markets, gamblers, mining pools, tumblers, and other services or markets. Gamblers, darknet markets, and tumblers are reminiscent of unlawful activities. Darknet markets are web pages with trade purposes in the Darknet [8]. They often deal drugs, cyber weapons, fake cash, stolen credit card details, forged documents, and other illegal stuff. Darknet is accessed with Tor as an intermediary [9]. A study by [10] in 2014 has shown that the second most popular website in Tor belongs to the darknet market [11]. After Silk Road, contemporary markets leverage the anonymity of the Darknet and bitcoin payments or similar crypto-currencies [12]. Tumblers are allegedly responsible for money laundering by mixing transactions. Considering the traceability of the Bitcoin transactions, finding the source address from a specific address is possible. Therefore, the mixing requester receives or sends payments with new untraceable transactions for a charge.

Our main objective is to classify Bitcoin addresses into seven roles in the Bitcoin network, consisting of four ordinary services and three criminal roles. The focus is on the model's

accuracy on the three criminal roles based on evaluation metrics.

The rest of this paper is organized as follows. In Section II, we elaborate on pertinent terms and concepts. Section III discusses the studies relevant to this work. In Section IV, we justify the necessity of this study and how it fills the gap among other similar works. In Section V, we describe our method, which consists of data gathering, feature extraction, and applying different supervised learning models. Section VI presents the models' results and compares them based on evaluative metrics. Finally, Section VII concludes our work.

II. Technical Background

Blockchain

Blockchain is a distributed data structure of all Bitcoin network transactions, and the transactions are kept in discrete updates called a block. Mining, a random process, produces blocks approximately every 10 minutes. Each block contains a proof-of-work with heavy computation. Proof-of-work adjusts the block intervals and protects the blockchain from rewriting the history attacks: an attacker must outdo proof-of-works of the latest blocks to replace the former propagated blocks since each block is connected to the later blocks probabilistically unchangeable [13].

Bitcoin transactions

Bitcoin transactions consist of some inputs and outputs. The outputs are primarily called unspent transaction output (UTXO), and they are indivisible amounts of the coin with a script determining the obligatory condition for spending. Each transaction gathers several older confirmed UTXOs and forms new UTXOs.

Evaluation metrics

To measure the effectiveness of our machine learning models, we require metrics. These metrics have specific mathematical formulas that depend on some arguments: we define c as an arbitrary class, TP_c as correctly predicted addresses in class c (True Positive), TN_c as correctly not predicted addresses in class c (True Negative), N as the number of all the addresses, n_c as the number of all the addresses originally in class c, and p_c as the number of all the addresses predicted in class c.

The metrics formulas are:

$$Accuracy = \frac{\sum_c TP_c + TN_c}{N} \tag{1}$$

$$Recall_c = \frac{TP_c}{n_c} \tag{2}$$

$$Precision_c = \frac{TP_c}{p_c} \tag{3}$$

$$f1_c = 2 \times \frac{Precision_c \times Recall_c}{Precision_c + Recall_c} \tag{4}$$

$$weighted\ average\ f1 = \frac{\sum_c f1_c \times n_c}{N} \tag{5}$$

III. Related Work

There were several attempts to trace Bitcoin addresses and networks. In [14], and [15], Bitcoin Network Graph was introduced, consisting of each address as a node and any transaction relation as an edge. In [16], several machine learning models were trained on the Bitcoin Network Graph, with XGBoost as the most accurate one. Building the Bitcoin Network Graph is costly and also not time efficient; testing or training any model on the graph is slow and resource-consuming. This work extracts approximately 150 features and applies decision tree, random forest, K-nearest neighbors (KNN), Extreme Gradient Boosting (XGBoost), and Multi-Layer Perceptron (MLP) models. Also, the labeled addresses were gathered from 2020 to mid-2021.

In [17], a graph similar to Bitcoin Network Graph was designed. With some primary features extracted from this graph and Ethereum blocks, this work accurately classified five major Ethereum network services based on the graph neural network (GNN) model.

In [18], a more complicated concept for Ethereum addresses was defined to find the analogy between different groups of addresses. This study extracted many features by modeling transactions with a social network, and despite the previous research, it utilized more diverse and valuable classes of addresses.

IV. Motivation

Bitcoin, the first cryptocurrency, has dedicated a large market capitalization to itself. According to [19], Bitcoin market value has reached 1.28 trillion dollars, gaining half of the whole cryptocurrency market. For profitable opportunities, Bitcoin has attracted users in various affairs: anonymous payments, investment, gambling, and money laundering.

Bitcoin anonymity has motivated illegal activities. However, it is semi-anonymous [20]. In other words, linking an address to its owner is quite difficult. Nevertheless, some clues reveal a connection between addresses and users or the addresses and their usage.

There are several ways to find these connections. The classification of the addresses based on transaction history happens to be an effective technique. Relevant studies have not considered the industrial aspect, while our work pondered this potential.

[16], [21], [22], and [23] employ Bitcoin Graph Network, and extract features from the graph. The computation on a colossal graph is complex and heavy. Also, the sustenance of this graph is challenging and resource-consuming. Although [24] and [25] avoid graphs and prefer direct feature extraction, they have fewer variant labels or accuracy in the best model. Our paper is advantageous in both aspects: Table I shows this comparison.

V. Method

In this section, data collection, saving and retrieving data, defining and extraction of features, and machine learning

	Our Work	[21]	[24]
Period	Jan 2020 - Mar 2020	3 Jan 2009 - 30 Jun 2018	2 May 2018 - 3 May 2018
Data Size	56,240	26,313	8,808
number of classes	7	7	6
Model	DT, KNN, XGB, RF	$AdaBoost, LightGB, LR, MLP, NN, RF, SVM, XGB$	DT, ET, NN, RF, SVM
Best Performance	0.98	0.87	0.96
Data Existence	Yes	No	Yes
Code Existence	Yes	No	No
Year	2022	2019	2020

TABLE I
COMPARISON OF OUR WORK WITH SIMILAR NONE GRAPH BASED METHODS

Class	Size
Centralized Exchange	31,252
Gambling	10,729
Cyber-Security Service	10,581
Tumbler	1,414
Darknet Market	1,226
P2P Financial Service	924
Mining Pool	114

TABLE II
THE DISTRIBUTION OF LABELED ADDRESSES CLASSES

models are presented. The programming resources are publicly available in the GitHub[1] repository.

A. Data Collection

There are two sets of data that we need to collect respectively, Bitcoin network transaction data and Bitcoin addresses with their corresponding labels. By bringing up a full node on the Bitcoin network and synchronizing it, we collected all the transactions of the Bitcoin network [26]. This data is divided into four files: header blocks, transactions, transaction inputs, and transaction outputs. 732,236 blocks and 726,277,179 transactions were gathered, equivalent to all blocks until April 17, 2022, with a volume of 700 GB. This considerable volume will involve long computations, so in this research, we use the blocks of the first two months of the year 2020. The reason for choosing this range is to use the latest addresses with enough activity; while very recent addresses may have little activity in the network, the training data might be of poor quality.

In contrast to transactions, gathering labeled addresses is more complicated. We applied address clustering methods, referred to in [27], to expand the labeled addresses from a primary set. Moreover, we used the WalletExplorer website that provides a user interface for wallet addresses [28]. Finally, 56,240 addresses were collected in 7 different classes. Table II contains the exact number of addresses tagged for each behavioral category. The imbalance is evident since, for some classes, such as the black market, collecting valid data is sophisticated.

B. Data sustenance and retrieval

Due to the high volume of transactions and the need to apply inter-file searches in transactions, join, and aggregation operations are required. If we intend to perform these operations in the RAM, the memory size must be several times the physical size of the data (about 70 Gigabytes). A computer with such technical specifications is unattainable; therefore, we had to utilize secondary storage. Due to the nature of the searches, a relational database is apt for our purpose. The most popular, optimal, and fastest of these systems is PostgreSQL [29]. In the database, four tables corresponding to each file were defined. The columns of each table correspond to the file's parameters. For faster query results, hash indexes were applied to target or joining columns, such as transaction id and address columns. To extract the features, inter-table and intra-table queries were employed.

[1]https://github.com/shayunak/Address-classification-project.git

C. Feature Selection

Features are divided into primary and secondary sets: primary features are directly extracted by the queries, while secondary ones result from dividing or ranging primary features. The queries contain aggregation or window functions for time-related features like transaction rate. Max, Min, Avg, Sum, and Std are aggregation functions utilized to build or expand primary features. Table III indicates the features, with "Aggregate Function" as the functions used to expand the features. Also, a line separates primary and secondary features in the table. Totally, the number of features turns 52.

D. Data Preparation

After feature computation, addresses might lack values for a feature. Based on our realization from Bitcoin transactions, the replacement is zero. Moreover, we split the data to train and test set by a 60-40 percent ratio, putting 0.6 for the "train_size" parameter of the "train_test_split" function in the "scikit-learn" python library. In addition, for hyper-parameter tuning, cross-validation with eight chunks is employed, putting 8 for the "cv" parameter of the "cross_val" function in the scikit-learn python library.

E. Model specifications

We use python "xgboost" and scikit-learn libraries to implement multi-class machine learning models. The hyper-parameter specifications are listed below:

- Decision Tree: DecisionTreeClassifier() from scikit-learn library with $random\_state = None$, $criterion =' entropy'$, $splitter =' best'$.

Name	Aggregate Function	Description
address_out_count	-	The number of outputs that an address can spend or has spent
spent	sum, max, min, avg, std	The amount of satoshi that an address can spend or has spent
address_in_count	-	The number of inputs that an address has spent
granted	sum, max, min, avg, std	The amount of satoshi that an address has spent
input_index	max, min, avg, std	The index of input in transactions that the address participated
output_index	max, min, avg, std	The index of output in transactions that the address participated
tx_cnt	-	The number of transactions that an address participated
first_time_tx	-	The height of the earliest block that an address appears in
last_time_tx	-	The height of the latest block that an address appears in
tx_size	max, min, avg, std	The size in bytes of the transaction that an address appears in
tx_num_of_inp	max, min, avg, std	The number of inputs of the transaction that an address appears in
tx_num_of_out	max, min, avg, std	The number of outputs of the transaction that an address appears in
interval	max, avg, std	The time(in blocks) between two consecutive transactions that an address appears in both
range_tx_num_of_out	-	The subtraction of tx_num_of_out_min from tx_num_of_out_max
range_tx_num_of_inp	-	The subtraction of tx_num_of_inp_min from tx_num_of_inp_max
tx_fee	-	The subtraction of sum_granted from sum_spent equivalent to the transaction fee
diff_in_out_count	-	The subtraction of address_in_count from address_out_count
ratio_in_out_count	-	The division of address_in_count by address_out_count
ratio_in_out_values	-	The division of sum_granted by sum_spent
range_spent	-	The subtraction of min_spent from max_spent
range_granted	-	The subtraction of min_granted from max_granted
ratio_range_to_value_output	-	The division of range_spent by sum_spent
ratio_range_to_value_input	-	The division of range_granted by sum_granted
range_output_index	-	The subtraction of max_output_index from min_output_index
range_input_index	-	The subtraction of max_input_index from min_input_index
life_period	-	The subtraction of first_time_tx from last_time_tx
range_tx_size	-	The subtraction of tx_size_min from tx_size_max

TABLE III
PRIMARY AND SECONDARY FEATURES

- Random Forest: RandomForestClassifier() from scikit-learn library with $random\_state = None$, $n\_estimators = 150$.
- K-Nearest Neighbors: KNeighborsClassifier() from scikit-learn library with $n\_neighbors = 5$, $algorithm =' kd\_tree'$, $weights =' distance'$.
- XGBoost: XGBClassifier() from xgboost library with $learning\_rate = 0.5$, $objective =' multi : softmax'$, $eval\_metric =' mlogloss'$

VI. RESULTS

The models get trained with fit() function and tested with predict() function. The evaluation to compare the model's performance is based on the parameters in Section II. Since gambling, darknet market, and tumbler classes are our targets, we include their f1-score in the result table, shown in Table IV.

According to the evaluation metrics, the performance of these four models can be sorted as follows:

$$KNN \ll RF \approx DT < XGB \tag{6}$$

XGBoost model outperforms the others. Therefore, in Fig 1, the confusion matrix for this model is shown for a detailed classification process. The main diagonal of the confusion matrix is populated as proof of the model's success.

Fig 2 displays the result of the feature importance process. The ratio of input-to-output participation turns out to be the most remarkable feature of the classification. As this feature is secondary, it demonstrates the effectiveness of the secondary feature notion and selection.

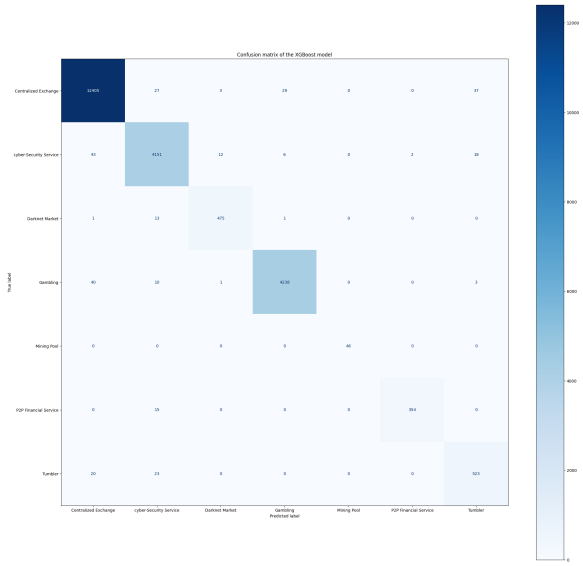

Fig. 1. XGBoost Confusion Matrix

VII. CONCLUSION AND FUTURE WORK

In this paper, we applied machine learning models to classify Bitcoin addresses into seven classes of ordinary roles and target roles with potential criminal backgrounds. These models function based on a 52-feature vector extracted from the transaction history on the blockchain. The best model turns out to be XGBoost with 98.7% weighted f1-score. According to the feature importance process, the ratio of output to input

Model	Gambling f1-score	Darknet f1-score	Tumbler f1-score	Weighted f1-score	Accuracy
Decision Tree	0.9492	0.9787	0.982	0.9716	0.9716
Random Forest	0.4528	1	0.9823	0.9731	0.9732
KNN	0.9492	0.3514	0.7532	0.697	0.7063
XGBoost	0.98	1	0.9895	**0.9865**	0.9865

TABLE IV
THE EVALUATION METRICS OF THE TESTING PHASE OF THE MODELS

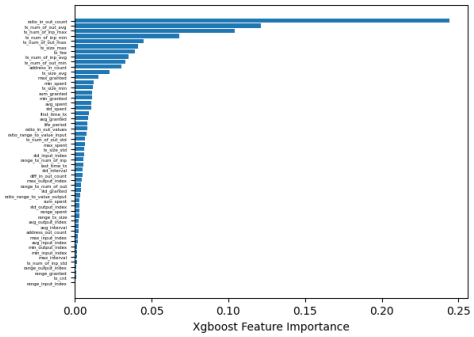

Fig. 2. XGBoost Feature Importance Chart

transactions was the most significant feature. Since it was a secondary feature, it proves the effectiveness of our feature selection. The model outperforms darknet market address classification, one of our target classes.

As future work, expanding the class diversity of labeled addresses and innovating new labeling methods can increase our model's accuracy. Also, adding geographical data inferred from the IP of the Bitcoin addresses to the feature set might improve the model's performance. Moreover, based on the feature importance process, removing weaker features makes the testing and training phases more efficient.

REFERENCES

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Business Review*, p. 21260, 2008.

[2] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[3] F. Wang, Y. Chen, R. Wang, A. O. Francis, B. Emmanuel, W. Zheng, and J. Chen, "An experimental investigation into the hash functions used in blockchains," *IEEE Transactions on Engineering Management*, vol. 67, no. 4, pp. 1404–1424, 2019.

[4] S. Haber and W. S. Stornetta, "How to time-stamp a digital document," in *Conference on the Theory and Application of Cryptography*. Springer, 1990, pp. 437–455.

[5] M. Apostolaki, C. Maire, and L. Vanbever, "Perimeter: A network-layer attack on the anonymity of cryptocurrencies," in *International Conference on Financial Cryptography and Data Security*. Springer, 2021, pp. 147–166.

[6] A. Biryukov, D. Khovratovich, and I. Pustogarov, "Deanonymisation of clients in bitcoin p2p network," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, 2014, pp. 15–29.

[7] P. Koshy, D. Koshy, and P. McDaniel, "An analysis of anonymity in bitcoin using p2p network traffic," in *International Conference on Financial Cryptography and Data Security*. Springer, 2014, pp. 469–485.

[8] P. Biddle, P. England, M. Peinado, B. Willman, *et al.*, "The darknet and the future of content distribution," in *ACM Workshop on digital rights management*, vol. 6, 2002, p. 54.

[9] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," Naval Research Lab Washington DC, Tech. Rep., 2004.

[10] "Patrick howell o'neill," https://www.technologyreview.com/author/patrick-howell-oneill/.

[11] M. Ward, "Tor's most visited hidden sites host child abuse images," *BBC News: Technology*, vol. 30, 2014.

[12] D. Gayle, "Online market'is turning drug dealers from goons to geeks," *The Guardian. Dostupno na:: https://www. theguardian. com/world/2016/feb/11/online-market-turning-drugdealers-goons-geeks-darknet*, 2016.

[13] G. O. Karame, E. Androulaki, and S. Capkun, "Two bitcoins at the price of one? double-spending attacks on fast payments in bitcoin," *Cryptology EPrint Archive*, 2012.

[14] I. Alqassem, I. Rahwan, and D. Svetinovic, "The anti-social system properties: Bitcoin network data analysis," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 1, pp. 21–31, 2018.

[15] B. Tao, I. W.-H. Ho, and H.-N. Dai, "Complex network analysis of the bitcoin blockchain network," in *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2021, pp. 1–5.

[16] Y. Xiang, Y. Lei, D. Bao, W. Ren, T. Li, Q. Yang, W. Liu, T. Zhu, and K.-K. R. Choo, "Babd: A bitcoin address behavior dataset for pattern analysis."

[17] J. Zhou, C. Hu, J. Chi, J. Wu, M. Shen, and Q. Xuan, "Behavior-aware account de-anonymization on ethereum interaction graph," *arXiv preprint arXiv:2203.09360*, 2022.

[18] G. Bonifazi, E. Corradini, D. Ursino, and L. Virgili, "Defining user spectra to classify ethereum users based on their behavior," *Journal of Big Data*, vol. 9, no. 1, pp. 1–39, 2022.

[19] "Bitcoin's market capitalization history," https://www.globaldata.com/data-insights/cards-amp-payments/bitcoins-market-capitalization-history/.

[20] M. Conti, E. S. Kumar, C. Lal, and S. Ruj, "A survey on security and privacy issues of bitcoin," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3416–3452, 2018.

[21] Y. Li, Y. Cai, H. Tian, G. Xue, and Z. Zheng, "Identifying illicit addresses in bitcoin network," in *International Conference on Blockchain and Trustworthy Systems*. Springer, 2020, pp. 99–111.

[22] 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*. Springer, 2017, pp. 248–263.

[23] 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," *arXiv preprint arXiv:1908.02591*, 2019.

[24] Y.-J. Lin, P.-W. Wu, C.-H. Hsu, I.-P. Tu, and S.-w. Liao, "An evaluation of bitcoin address classification based on transaction history summarization," in *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. IEEE, 2019, pp. 302–310.

[25] R. Michalski, D. Dziubałtowska, and P. Macek, "Revealing the character of nodes in a blockchain with supervised learning," *Ieee Access*, vol. 8, pp. 109 639–109 647, 2020.

[26] "Running a full node," https://bitcoin.org/en/full-node.

[27] M. Möser and A. Narayanan, "Resurrecting address clustering in bitcoin," in *International Conference on Financial Cryptography and Data Security*. Springer, 2022, pp. 386–403.

[28] "Walletexplorer.com: smart bitcoin block explorer," https://www.walletexplorer.com/.

[29] "Postgresql: The world's most advanced open source relational database," https://www.postgresql.org/.