Supervised Learning model for Identifying illegal activities in Bitcoin

Pranav Nerurkar, Sunil Bhirud, Dhiren Patel, Romaric Ludinard, Yann Busnel, Saru Kumari

Preprint to Applied Intelligence manuscript No.
(will be inserted by the editor)

Supervised Learning model for Identifying illegal activities in Bitcoin

Pranav Nerurkar · Sunil Bhirud · Dhiren Patel · Romaric Ludinard · Yann Busnel · Saru Kumari

Received: June 2020 / Revised: October 2020 / Accepted: November 2020

Abstract Since its inception in 2009, Bitcoin is mired in controversies for providing a haven for illegal activities. Several types of illicit users hide behind the blanket of anonymity. Uncovering these entities is key for forensic investigations. Current methods utilize machine learning for identifying these illicit entities. However, the existing approaches only focus on a limited category of illicit users. The current paper proposes to address the issue by implementing an ensemble of decision trees for supervised learning. More parameters allow the ensemble model to learn discriminating features that can categorize multiple groups of illicit users from licit users. To evaluate the model, a dataset of 1216 real-life entities on Bitcoin was extracted from the Blockchain. Nine Features were engineered to train the model for segregating 16 different licit-illicit categories of users. The proposed model provided a reliable tool for forensic study. Empirical evaluation of the proposed model vis-a-vis three existing benchmark models was performed to highlight its efficacy. Experiments showed that the specificity and sensitivity of the proposed model were comparable to other models. Due to higher parameters of the ensemble tree model, the classification accuracy was 0.91, with 95% CI - 0.8727, 0.9477. This was better than SVM and Logistic Regression, the two popular models in the literature and comparable to the Random Forest and XGBOOST model. CPU and RAM utilization were also monitored to demonstrate the usefulness of the proposed work for real-world deployment. RAM utilization for the proposed model was higher by 30-45% compared to the other three models. Hence, the proposed model is resource-intensive as it has higher parameters than the other three models. Higher parameters also result in higher accuracy of predictions.

Keywords Bitcoin · Fraud detection · Exploratory Data Analysis

P. Nerurkar
Dept. of Data Science, NMIMS and Dept. of CE&IT, VJTI Mumbai
E-mail: panerurkar_p16@ce.vjti.ac.in / pranav.n@nmims.edu

Sunil Bhirud, Dhiren Patel
Dept. of CE&IT, VJTI Mumbai

R. Ludinard, Y. Busnel
SRCD Department, IMT Atlantique, IRISA, Rennes, France

S. Kumari
Dept. of Mathematics, Ch. Charan Singh University, Meerut, India
E-mail: saryusiirohi@gmail.com

1 Introduction

Bitcoin [1] platform has attracted both social and anti-social elements [41, 11, 7, 18, 45]. On the one hand, it is social as it ensures the exchange of value, maintaining trust in a cooperative, community-driven manner without the need for a commissioned third party. At the same time, it is anti-social as it creates hurdles for law enforcement to trace suspicious transactions due to the anonymity and privacy [39, 31, 34]. Since Bitcoin's inception in 2009, the initial two years saw slow adoption with hardly 1000 unique addresses and less than 10000 transactions per day [34]. However, as Bitcoin became financially significant, there was an exponential growth in transactions from 2012-2016, which also saw the entry of serious users viz. mixing services [9], gambling sites, trading exchanges, investors, speculators, and independent mining industries [17]. The change in the profile of Bitcoin's user base was also evident from the increase in the transaction values, fluctuations in BTC price, and BTC's volume. The 2012-onwards phase saw the emergence of Ponzi schemes, money laundering, frauds [8], embezzlements, extortion [40, 33] and tax evasion [44] practices that used the blanket of secrecy afforded by Bitcoin to mislead the audit trail. It was speculated that in 2017, BTCs worth $770 million were traded for illicit activities [22], a quarter of Bitcoin users were malicious and 46% of all Bitcoin activity was illegal [13].

Due to voluminous data about Bitcoin transactions on the blockchain, machine learning became a popular technique for tracking and scrutinizing illicit users or transactions. Existing literature surveyed on detecting illegal activities using Machine Learning (ML) had focused on deanonymizing entities [24, 55, 42, 20], detecting botnets [52], illegal transactions [22], identifying suspicious Bitcoin users [48, 49, 47, 43, 50, 54, 18, 45] (extortionists [36], ponzi scams [4], darknet markets [21], ransomwares [2], human traffickers [38], frauds [29, 30]), detecting money laundering [17, 51, 15], identifying mixing services [32], identifying Bitcoin exchanges [23], identifying illegal transactions [35, 6], identifying Bitcoin wallets [1] and Bitcoin miners [53]. The standard pipeline followed by these studies is given in Figure 1.

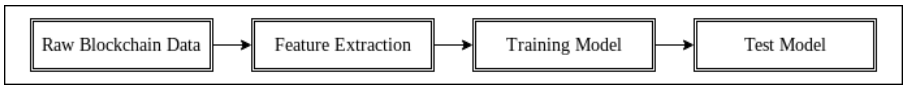


Fig. 1: Pipeline of ML on Bitcoin system

Scope for feature engineering and extraction is immense due to the vast categories of metadata associated with Blockchain (see Figure 2). Machine learning or deep learning has brought about paradigm shifts in modeling entities in domains such as image recognition, object localization, or audio or

[1] In this paper, Bitcoin refers to the system, and bitcoin or BTC refers to the digital currency

speech processing. However, as cryptocurrencies are still in their nascent stages, machine learning has made limited progress. The issues faced in the application of machine learning in identifying illegal activities are lack of benchmark, public datasets (see Table 3) [42], full information of Blockchain, and lack of ground truth information on the identities of Bitcoin users. Apart from these issues, cryptocurrencies offer their users pseudo-anonymity by allowing users to transact with each other through hash address. These addresses can be created and discarded countless times, complicating the task of linking a transaction to a user. Existing studies used deanonymizing techniques (see Section 2.3) to link multiple hash addresses to a single entity. To reduce the computational complexity of machine learning models, the target of interest was restricted to limited categories of illicit users. Additionally, the time interval for which data was collected from the Blockchain for feature engineering was restricted to shorter spans. Due to these, the models obtained after training were not generalized.

1.1 Motivation

The current paper aimed to build upon and extend work in detecting illegal entities in Bitcoin. Features of Bitcoin users were derived by scrutinizing the Blockchain from 03 Jan 2009 12:45:05 GMT to 08 May 2020 at 13:21:33 GMT. This was to provide the model with features suitable for generalized learning of entity behavior. Additionally, this avoided a model trained for recognizing a limited category of entities.

1.2 Contributions

The following contributions are proposed in the paper:

- General comprehension of machine learning techniques for recognizing malicious users in the Bitcoin network;
- A public dataset of addresses and features of illicit Bitcoin entities [2]
- A public repository of scripts for extraction of features of entities from Bitcoin blockchain, and associating hash addresses with entities [3]
- Empirical analysis of different learning strategies for classifying illicit Bitcoin entities;
- Implementing a supervised learning approach that estimated the most discriminating features for detecting categories of illicit Bitcoin users.

1.3 Novelty

An extensive literature survey could find studies focusing on only a subset of illicit activities viz. botnets, extortionists, ponzi scams, darknet markets,

[2] https://drive.google.com/open?id=1YdPj8whgbCKORuW3E0rhgfQIHkcFj9S5
[3] https://www.kaggle.com/pranavn91/blockchain

ransomware, human traffickers, frauds, money laundering, and mixing services. At the time of writing (June 2020), there has not been any research focusing on a broad spectrum of illegal activities.

1.4 Outline

The rest of the paper is organized into four sections. Section 2 provides the preliminaries needed for the paper, along with a critique of the current literature. Materials and methods detail the data collection and preparation strategy in Section 3. The proposed work is described in Section 4 followed by Experimental study in Section 5 and Conclusion and future works in Section 6.

2 Related work

Anatomy of Bitcoin and fundamental concepts such as blocks, Blockchain, transactions, inputs, outputs, current services on Bitcoin, deanonymization are described in Sections 2.1, 2.2, and 2.3. Followed by critical analysis of published studies on detecting illegal users (see Section 2.4), issues in available datasets (see Section 2.5) and popular ML models used in published studies (see Section 2.6).

2.1 Description of Bitcoin system

Bitcoin transactions are added to "Blocks" and recorded into a distributed public ledger "Blockchain". Each transaction has several inputs (senders) and outputs (receivers). The metadata [4] associated with blocks, transactions, inputs and outputs provides scope for analysis (see Figure 2). A single Bitcoin user can generate multiple addresses for sending and receiving BTCs, which creates a disadvantage in scrutinizing Bitcoin users. Deanonymizing techniques provide a solution to overcome this problem.

2.2 Common types of services on Bitcoin

The following types of services operate on the Bitcoin network:

- Exchanges (E): Allow trading of BTC to fiat currencies
- Pools (P): Individual users combine their processing power for mining blocks
- Gambling (G): Allow placing of bets using BTCs
- Wallets (W): Store BTC private keys and balance
- Payment gateways (PG): Allow accepting payment for services in BTCs

[4] https://github.com/blockchain-etl

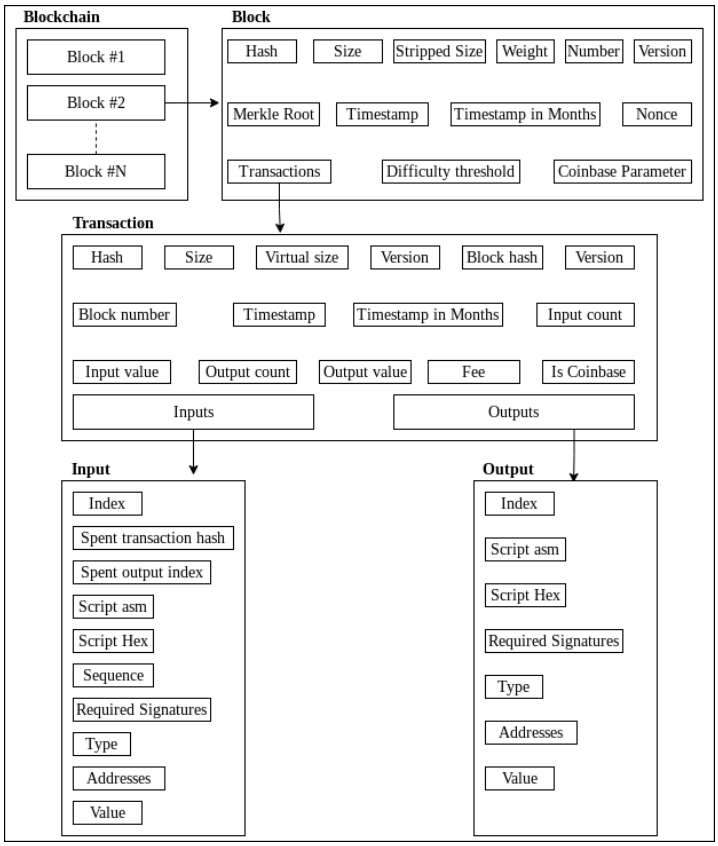


Fig. 2: Anatomy of Bitcoin system

- Miner (M): Organizations competing to mine blocks
- Darknet markets (DM): Selling and buying goods using BTCs
- Mixers (MX): Remove traceability of BTCs from source
- Trading sites (T): Purchase equities using BTCs
- P2Plenders (P2P): Crowdsourcing BTCs for loans
- Faucets (F): Reward in BTCs to subscribers
- Explorer (EX): Educational websites provide API to explore Bitcoin
- P2PMarket (P2PM): Marketplace for second-hand goods where buyers can contact sellers, payments in BTCs
- Bond markets (B): Buying bonds or debt instruments in BTC
- Affiliate marketers (AM): Pay per click in BTC
- Video sharing (VM): Payment in BTCs for viewing videos
- Money launderers (ML): Convert fiat currencies to BTC
- Cyber-security providers (CSP): Provide cybersecurity products for BTC
- Cyber-criminals (CC): Blacklisted by governments
- Ponzi (PZ): High yield investment scams

2.3 Bitcoin and Deanonymization

Block is a set of transactions $T = \{t_1, t_2, ..., t_n\}$. For every $t_i \subset T$ there is a 3-tuple (t_s, I^{t_i}, O^{t_i}) where t_s denotes UNIX timestamp of t_i and I, O denotes the addresses of inputs (senders) and outputs (receivers) in t_i respectively [52]. Each t_i can have several inputs and outputs i.e., $I^{t_i} = \{i_1, i_2, ..., i_n\}$ and $O^{t_i} = \{o_1, o_2, ..., o_n\}$. Each Bitcoin user $u_i \subset U$ where $U = \{u_1, u_2, u_3, ..., u_n\}$ can have multiple addresses and perform multiple transactions. For sending bitcoins (BTCs), u_i can generate a new address for each transaction t_i.

The task of a deanonymizing function $f(.)$ is combining all addresses generated by u_i i.e., $A^{u_i} = \{{i_{t_1}}^{u_i}, {i_{t_2}}^{u_i}, ..., {i_{t_n}}^{u_i}, {o_{t_1}}^{u_i}, {o_{t_2}}^{u_i}, ..., {o_{t_n}}^{u_i}\}$, across all transactions. Here ${i_{t_1}}^{u_i}$ is address generated by u_i to send BTCs in t_1 and ${o_{t_n}}^{u_i}$ is address generated by u_i to receive BTCs in t_n.

Deanonymizing is a non-trivial task due to the complexity and diversity of the Bitcoin network [16, 14]. Functions proposed in the literature can be categorized as heuristic-based [44, 37, 31, 52, 5], distributed network-based [12] and machine learning-based [24]. Heuristic-based functions are the most popular and widely used in Bitcoin studies.

2.4 Studies on detecting illegal activities in Bitcoin

An advantage in the study of crypto-currencies is that transaction records are maintained on a distributed ledger "Blockchain", which is openly available for examination. The volume of the Blockchain presents problems in scrutinizing it, limiting the timespan of study, or restricting the objectives were used by studies in the literature to overcome this issue.

Literature surveyed on detecting illegal activities has focused on deanonymizing entities [24, 55, 42, 20], detecting botnets [52], illegal transactions [22], identifying suspicious Bitcoin users [48, 49, 47, 43, 50, 54, 18, 45] (extortionists [36], ponzi scams [4], darknet markets [21], ransomwares [2], human traffickers [38], frauds [29, 30]), detect money laundering [17, 51, 15], identifying mixing services [32], identify Bitcoin exchanges [23], identify illegal transactions [35, 6], identifying Bitcoin wallets [1] and Bitcoin miners [53]. Table 1 summarizes the strategies used in these studies.

Feature engineering is the most critical aspect of Bitcoin-based studies focusing on illicit activity or illicit user detection. Various approaches used by the authors for feature engineering can be grouped into five types (see Table 2).

Table 1: Summary of published Bitcoin studies

Authors	**Description**	**Features extracted**
B Zarpelao *et al.* [52]	Detection of botnets using Bitcoin protocols to launch DDoS attacks	Transaction features
T Liu *et al.* [24]	Deanonymize Bitcoin address	Network based features
C Lee *et al.* [22]	Detecting Illegal Transactions on Bitcoin	Transaction features
Y Wu *et al.* [48, 49]	Tracing suspicious Bitcoin entities	Transaction features
M Weber *et al.* [47]	Identifying illicit Bitcoin users	Transaction features
Y Hu *et al.* [17, 51, 15]	Detecting Money Laundering Activities	Graph embeddings
H Yin *et al.* [43]	Identifying illicit Bitcoin users	Transaction features
L Nan *et al.* [32]	Mixing service detection	Graph embeddings
L Yang *et al.* [50]	Identifying illicit Bitcoin users	Transaction features
J Liang *et al.* [23]	Bitcoin Exchange Identification	Graph embeddings
Z Zhang *et al.* [54]	Identifying illicit Bitcoin users	Transaction features
T Pham *et al.* [35]	Detecting Illegal Transactions on Bitcoin	Clustering nodes based on transaction features
A Bogner [6]	Detecting Illegal Transactions on Bitcoin	Clustering nodes based on transaction features
F Zola *et al.* [55]	Deanonymize Bitcoin address	Transaction features
F Aiolli *et al.* [1]	Identifying Bitcoin wallets	Transaction features
W Shao *et al.* [42]	Deanonymize Bitcoin address	Transaction features
M Vasek *et al.* [46]	Identifying Bitcoin scams	Transaction and network features
M Bartoletti *et al.* [4]	Identifying Bitcoin ponzi schemes	Transaction and network features
P Monamo *et al.* [29, 30]	Identifying Bitcoin fraud schemes	Clustering nodes based on transaction features
J Munoz [53]	Identifying Bitcoin miners	Network traffic features
A Irwin *et al.* [18, 45]	Identifying illicit Bitcoin users	Transaction features
R Portnoff *et al.* [38]	Identifying human traffickers in Bitcoin	Transaction features
C Ackora *et al.* [2]	Identifying ransomware in Bitcoin	Transaction features
K Kanemura *et al.* [21]	Identifying darknet markets in Bitcoin	Transaction features
M Jordan *et al.* [20]	Deanonymize Bitcoin address	Transaction features
S Phetsouvanh, *et al.* [36]	Identifying extortionists in Bitcoin	Transaction and network features

Table 2: Types of features used in published Bitcoin studies

Types of features	**Description**
Transaction	Total inputs, Total outputs, Total amount sent/received, Average amount sent/received, Standard deviation of amount sent/received, Time interval between successive transactions, Wallets transacted with, Number of addresses of an entity, BTCs sent, BTCs received, USD value of transactions, Timestamp, Wallet balance, wallet creation date, wallet active duration, Difference in wallet balance between successive days, IP address
Network	In-degree, out-degree, unique in-degree, unique out-degree, clustering coefficient, Gini coefficient, Number of triangles formed, measures of betweenness centrality, closeness centrality, degree centrality, in-degree centrality, out-degree centrality, PageRank, and load centrality.
Graph embeddings	RandomWalk, Node2Vec, DeepWalk, GCN, EvolveGCN, Structural deep network embedding (SDNE), Deepneural networks for learning graph representations (DNGR)
Clustering	KMeans, DBSCAN, AGNES, DIANA
Network traffic	Packets set/received per second, Average bits per packet, Amount of packets per second sent and received each second for each coin, average number of bits each packets holds in each flow, sent and received for each coin

2.5 Datasets used in published Bitcoin studies

The availability of standard datasets is a critical issue in examining Bitcoin. The entire Blockchain from inception to 08 May 2020 at 13:21:33 GMT was 298GB. Due to storage, computational, and time complexity, majority researchers (excluding surveys [28, 27, 26, 25, 3, 11]) have focused on limited categories of illicit users and shorter periods.

Table 3: Datasets used in published Bitcoin studies

Dataset	**Accessibility**	**Features**	**Categories**	**Size**
Chainanalysis [15, 43, 51]	Private	9	exchange, gambling, hosted wallet, merchant services, miningpool, mixing, ransomware, scam, tor market or other	198,097,356
Univ. Illinois Urbana-Champaign [35, 29, 30]	Public	0	0	37,450,461
BitcoinPonzi [4]	Public	11	Ponzi, Non-ponzi	6432
R Portnoff *et al.* [38]	Private	2	Sex offender, Ordinary	753,929
D Ermilov *et al.* [12]	Private	238	Service, gambling, mixer, exchange, pool, darknet	244,030,115
Ellipse [47]	Public	6	licit, illicit	203,769
C Lee *et al.* [22]	Private	2	licit, illicit	2 million
M Vasek *et al.* [46]	Private	2	Ponzi schemes, mining scams, scam wallets and fraudulent exchange	192
Wei Shao *et al.* [42]	Private	173	NA	10000

2.6 Role of ML models in published Bitcoin studies

Table 4 gives the popular ML models for Bitcoin studies.

Table 4: ML classifier used in published Bitcoin studies

ML models	**Research Paper**
k-Nearest Neighbours	[15, 43, 51]
Random Forests (RF)	[15, 43, 51, 4, 55, 1, 23]
Extra Trees	[15, 43, 51]
Decision Trees	[15, 43, 51, 23]
Bagging Classifier	[15, 43, 51, 53]
Gradient Boosting	[15, 43, 51, 55, 2]
AdaBoost	[15, 43, 51, 55]
Support Vector Machine (SVM)	[51, 35, 1, 23]
MultiLayer Perceptron	[51, 4, 22, 42, 23]
K-Means	[35, 30, 29]
Graph convolutional network	[47]
Logistic regression (LogReg)	[23]
DeepWalk, Node2vec, SDNE	[17, 32]

From Table 1 in literature, it is clearly observed that the implementation of a reliable and secure illegal user detection system is a primary concern for privacy and security in Bitcoin. Existing works have not focused on a broad spectrum of illegal activities that are conducted on Bitcoin. Additionally, existing datasets are unsuitable for machine learning, as their focus is narrow or are proprietary. In this respect, in Section 3, describes the data collection methodology to overcome the issue of data availability in public datasets. Section 4 discusses the proposed methodology for a classifier that could identify a broad spectrum of illicit users on Bitcoin.

3 Materials and Methods

As available datasets in literature (see Table 3) have shortcomings, procedure described in Section 3.1 was used to extract features mentioned in Section 3.2 of Bitcoin entities. Hardware and software configuration used for data collection is given in Section 3.3.

3.1 Data collection and preprocessing

Bitcoin blockchain dataset in raw form was obtained from full node at VJTI Blockchain lab [5]. The dataset was of size 298GB and consisted of Blockchain in the form of blk.data files. All blocks and transactions from 03 Jan 2009 12:45:05 GMT to 08 May 2020 13:21:33 GMT were present in the dataset. This raw data was then converted to CSV files using the blockchain parser built by the VJTI Blockchain lab [6]. The processed dataset, is made available for download [7]. Table 5 shows the three ".csv" files of the processed dataset.

Table 5: Description of processed dataset

Relation	**Attributes**		
Output	tx_hash:ID	receiver_address	amount
Inputs	sender_address	tx_hash:ID	amount
Transactions	tx_hash:ID	timestamp	

From the Transactions dataset, it is possible to obtain the count of transactions occurring in that year. Each transaction (tx) was identified in Blockchain by a unique hash (tx_hash:ID) and had a timestamp, the UNIX time of the transaction. For the year 2009, transactions start from 03 Jan 2009 12:45:05 GMT, and for the year 2020, transactions up to 08 May 2020 at 13:21:33 GMT were considered. Table 6 and 7 describe the growth observed on various measures in Bitcoin dataset during 2009-2020.

[5] https://www.vjti-bct.in/

[6] https://github.com/pranavn91/blockchain

[7] https://drive.google.com/open?id=1pEpBAUXKgQX0BP8ircQgd9yXiucLY14h

Table 6: Distribution of transactions in Bitcoin blockchain network (2009-2015)

	2009	2010	2011	2012	2013	2014	2015
Transactions	32741	185410	1902443	8459093	19645798	25265702	45689861
Inputs	2810	108965	1902443	5716084	15407017	33300547	54564769
Outputs	32643	143863	2595309	5981241	16278420	34586691	57150816
Max BTC's in a tx	22500	96999	550000	158336.30	194993.50	217517.63	172841.81
Max inputs in a tx	320	901	529	673	1757	674	1519
Max outputs in a tx	2	98	2002	2792	3075	5352	13107
Input sending highest amount	COINBASE	COINBASE	CoinJoin Mess	DeepBit.net	DeepBit.net	Unknown	Unknown
Output receiving highest amount	Unknown	Unknown	CoinJoin Mess	DeepBit.net	DeepBit.net	Unknown	Unknown
Total BTCs sent	1978736	22667790	297984085	925215501	429732306	264107039	548006072

Table 7: Distribution of transactions in Bitcoin blockchain network (2016-2020)

	2016	2017	2018	2019	2020
Transactions	82634637	104081930	81393458	119729415	39978670
Inputs	90773554	128642149	77568478	128768057	52805351
Outputs	95783964	144361281	104780607	133558733	54179450
Max BTCs in a tx	99489.99	87082.81	109735.6	157457.612	182501
Max inputs in a tx	677	1089	1061	1347	1442
Max outputs in a tx	11515	6626	5027	7266	6990
Input sending highest amount	Unknown	Unknown	Unknown	Unknown	Unknown
Output receiving highest amount	Unknown	Unknown	Unknown	Unknown	Unknown
Total BTCs sent	1068404725	896026050.66	290858051.91	515972850.159	128637285.824

3.2 Feature extraction

Based on the structure of Bitcoin (see Figure 2), features to train the classifier were extracted from Blockchain data to build a dataset for training the classifier. Feature list is given in Table 8.

Table 8: List of Features

Feature symbol	**Feature description**
T_x	Total transactions in which wallet has participated
B	Current BTC present in the wallet
${T_x}^{in}$	Total incoming transactions to the wallet
${T_x}^{out}$	Total outgoing transactions from the wallet
L	Total active life of the wallet
A_w	Total addresses of the wallet
A_v	Average number of incoming transactions received by an address of a wallet
T	Total number of addresses sending BTC to the wallet
R	Ratio of Transaction count and address count. Gives the average number of times an address of the wallet was reused for a transaction.

Multiple hash addresses belonging to a single entity were clustered using multi-input heuristic clustering [28, 27, 26, 25]. Features extracted for each entity would allow a classifier to learn the categorization of each wallet into one of the nineteen categories (see Section 2.2). Bitcoin entities were

identified using an API [8] [19]. The API helped in building a labeled dataset for supervised learning (see Table 9) with 1216 observations. Figure 3 illustrates the flowchart for the proposed work.

Table 9: Types of Bitcoin entities in dataset

affiliatemarketing	**criminals**	**cybersec**	**darkmarket**
2	1	2	15
exchange	**faucet**	**gambling**	**miner**
78	2	24	2
mixer	**p2plender**	**paymentgateway**	**ponzi**
62	5	6	19
pools	**trading**	**Unclassified**	**wallets**
20	2	968	8

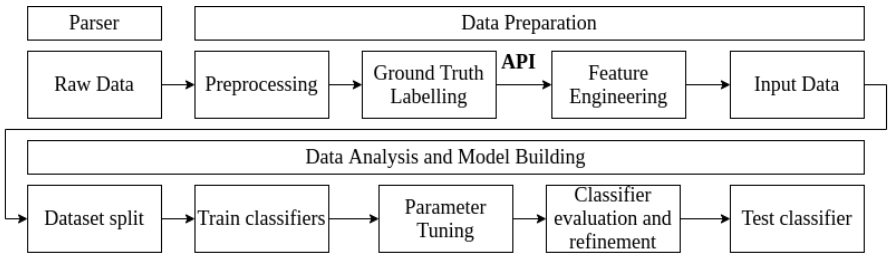


Fig. 3: Flowchart of proposed work

3.3 Experimental setup

The preprocessing code was in Python 3.6, and the experiments were performed on a single core 1 TB Intel(R) Xeon(R) Silver 4114 CPU@2.20GHz. API calls were made through the Curl package of R using Jupyter notebooks hosted on Google Colab (n1-highmem-2 instance, 2vCPU @2.20GHz, 13GB RAM) and Kaggle (Intel(R) Xeon(R) CPU @2.20GHz, 13GB RAM).

4 Classification of Illicit entities in Bitcoin

The mathematical model of the proposed classifier is given in Section 4.1 with steps used for training it listed in Section 4.2. For the implementation of gradient boosting, the R Caret package was used. It is different from XGBOOST package [10] as it does not use column sub-sampling, cache-aware

[8] https://github.com/pranavn91/blockchain/blob/master/walletexplorer-api

access, sparsity split finding and parallel computation. Features of XGBOOST make computation slower with limited improvement in accuracy.

4.1 Mathematical model

Each training sample $(x^{(i)}, y^{(i)})$ of dataset $\mathcal{D}$ is with m features and total n samples are present in the dataset. Hence, $\mathcal{D} = \{(x_i, y_i)\}$ $(|\mathcal{D}| = n, x_i \in \mathbb{R}^m, y_i \in \mathbb{R})$. The proposed tree ensemble model uses K additive functions to predict the output.

$$\hat{y}_i = \phi(\mathbf{x}_i) = \sum_{k=1}^{K} f_k(\mathbf{x}_i), \quad f_k \in \mathcal{F}, \tag{1}$$

where

- $\mathcal{F} = \{f(\mathbf{x}) = w_{q(\mathbf{x})}\}(q : \mathbb{R}^m \to T, w \in \mathbb{R}^T)$: set of regression trees
- q: tree structure mapping an $x^{(i)}$ to its leaf index
- T: leaves count in the tree
- f_k: q having leaf weights w
- w_i: score on i^{th} leaf

For each $x^{(i)}$ the decision rules of q classify it into the leaf nodes and calculate the final prediction by $\sum w$ i.e. summing up the score in the corresponding leaves. The obtain the optimal model ϕ, the loss $\mathcal{L}(\phi)$ is minimized by following *regularized* objective.

$$\begin{aligned} \mathcal{L}(\phi) &= \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k) \\ \text{where } \Omega(f) &= \gamma T + \frac{1}{2}\lambda \|w\|^2 \end{aligned} \tag{2}$$

where,

- l: loss function
- $\hat{y}_i$: prediction
- y_i: target
- Ω: regularization term

4.1.1 Optimization

As the loss function given in Eq. 2 cannot be optimized using standard optimization techniques, the model is trained in an additive manner specified in [10]. Eq. 2 is modified as Eq 3.

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y_i}^{(t-1)} + f_t(\mathbf{x}_i)) + \Omega(f_t) \tag{3}$$

where,

– $\hat{y}_i^{(t)}$: prediction of the i-th instance at the t-th iteration
– f_t: q having leaf weights w

Using second-order approximation for optimizing Eq 3,

$$\mathcal{L}^{(t)} \simeq \sum_{i=1}^{n}[l(y_i, \hat{y}^{(t-1)}) + g_i f_t(\mathbf{x}_i) + \frac{1}{2}h_i f_t^2(\mathbf{x}_i)] + \Omega(f_t)$$

where $g_i = \partial_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})$ and $h_i = \partial^2_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})$ are first and second order derivatives on the loss function.

Removing the constant terms in Eq 4.1.1.

$$\tilde{\mathcal{L}}^{(t)} = \sum_{i=1}^{n}[g_i f_t(\mathbf{x}_i) + \frac{1}{2}h_i f_t^2(\mathbf{x}_i)] + \Omega(f_t) \tag{4}$$

By defining $I_j = \{i|q(\mathbf{x}_i) = j\}$ to be instance set of leaf j and expanding Ω, rewriting Eq 4 as Eq 5

$$\begin{aligned}\tilde{\mathcal{L}}^{(t)} &= \sum_{i=1}^{n}[g_i f_t(\mathbf{x}_i) + \frac{1}{2}h_i f_t^2(\mathbf{x}_i)] + \gamma T + \frac{1}{2}\lambda\sum_{j=1}^{T} w_j^2 \\ &= \sum_{j=1}^{T}[(\sum_{i\in I_j} g_i)w_j + \frac{1}{2}(\sum_{i\in I_j} h_i + \lambda)w_j^2] + \gamma T\end{aligned} \tag{5}$$

After obtaining $q(\mathbf{x})$ computation of optimal weights w_j^* of leaf j by Eq 6

$$w_j^* = -\frac{\sum_{i\in I_j} g_i}{\sum_{i\in I_j} h_i + \lambda}, \tag{6}$$

leads to optimal value as Eq 7

$$\tilde{\mathcal{L}}^{(t)}(q) = -\frac{1}{2}\sum_{j=1}^{T}\frac{(\sum_{i\in I_j} g_i)^2}{\sum_{i\in I_j} h_i + \lambda} + \gamma T. \tag{7}$$

The proposed model uses Eq 7 to measure quality (impurity score) of q. A greedy algorithm to estimate the optimal q, initiates the tree from a single leaf node by iteratively adding branches to it. Given the leaves in left and right nodes, loss reduction after the split is given by Eq 8,

$$\mathcal{L}_{split} = \frac{1}{2}\left[\frac{(\sum_{i\in I_L} g_i)^2}{\sum_{i\in I_L} h_i + \lambda} + \frac{(\sum_{i\in I_R} g_i)^2}{\sum_{i\in I_R} h_i + \lambda} - \frac{(\sum_{i\in I} g_i)^2}{\sum_{i\in I} h_i + \lambda}\right] - \gamma \tag{8}$$

where,

– I_L and I_R: instance sets of left and right nodes after the split
– $\mathcal{L}_{split}$: loss reduction after the split

4.2 Steps for training proposed model

Algorithm 1 Classifier training, parameter tuning, refinement and evaluation

```
Set initial seed for random numbers
Set the training control values
Set the tuning grid for parameter search
for each parameter set do
    for each resampling iteration set do
        Hold out specific samples
        Pre process the data (Center and Scale)
        Fit the model on the remaining samples
        Predict the held out samples
    end for
    Calculate the average performance across held out predictions
end for
Determine the optimal parameter set
Fit the final model to all the training data using the optimal parameter set
```

4.3 Time complexity

The proposed model calculates the quality function (Eq. 7) based on each split of the Bitcoin data which has m features and total n samples, and it does this for each feature in every node that is not a leaf node. The best case of a balanced tree, the depth would be in $\mathcal{O}(logn)$; however, the decision tree performs locally optimal splits. Thus the worst case of depth being in $\mathcal{O}(n)$ is possible - if each split results in 1 and $f-1$ examples, where f is the number of examples of the current node. Hence, the time complexity for decision trees is within $\mathcal{O}(nmlogn)$ and $\mathcal{O}(n^2m)$. The uncertainty in depth is due to the non-deterministic way in which decision trees are built.

5 Experimental study

The proposed model in Section 4 was evaluated on dataset described in Table 9 for the experimental study with metrics (Section 5.2).

5.1 Description of experiment

Comparative study was performed of popular non-parametric ML models in literature - SVM, LogReg, XGBOOST, RF (see Section 4) with proposed model for evaluating classification accuracy on dataset (Table 9). The hyper-parameters for the baseline models are given.

- SVM: degree = 1, scale = 0.1274557 and C (cost) = 150.1363
- LogReg: cost = 154.3669, loss = L1 and epsilon (tolerance) = 1.

- XGBOOST: base_score=1, booster=1, colsample_bylevel=1, colsample_bynode=1, colsample_bytree=1, gamma=0.1, gpu_id=1, importance_type='gain', interaction_constraints=2, learning_rate=0.001, max_delta_step=0.1, max_depth=3, min_child_weight=2, missing=nan, monotone_constraints=1, n_estimators=100, n_jobs=-1, num_parallel_tree=1, objective='binary:logistic', random_state=7, reg_alpha=0.001, reg_lambda=0.01, scale_pos_weight=0.5, subsample=True, tree_method='ada', validate_parameters=True, verbosity=0
- RF: Randomly Selected Predictors = 1
- Proposed model: Boosting Iterations = 694, Max Tree Depth = 10, Shrinkage = 0.349461, Minimum Loss Reduction = 3.100079, Subsample Ratio of Columns = 0.3621352, Minimum Sum of Instance Weight = 1 and Subsample Percentage = 0.7216631

5.2 Metrics

Given the true positives t_p, true negatives t_n, type I error f_p and type II error f_n obtained from observing $(\hat{y}_i, y_i)$, following metrics were used.

$$Sensitivity(S) = \frac{t_p}{t_p + f_n} \tag{9}$$

$$Specificity(S_p) = \frac{t_n}{t_n + f_p} \tag{10}$$

$$Accuracy(A) = \frac{t_p + t_n}{t_p + t_n + f_p + f_n} \tag{11}$$

$$Prevalence(P) = \frac{t_p + f_n}{t_p + t_n + f_p + f_n} \tag{12}$$

$$PositivePredictionValue(PPV) = \frac{S * P}{((S * P) + ((1 - S_p) * (1 - P))} \tag{13}$$

$$NegativePredictionValue(NPV) = \frac{S * (1 - P)}{((1 - S) * P) + ((S_p) * (1 - P))} \tag{14}$$

$$Detectionrate = \frac{t_p}{t_p + t_n + f_p + f_n} \tag{15}$$

$$Detectionprevalence = \frac{t_p + f_p}{t_p + t_n + f_p + f_n} \tag{16}$$

$$BalancedAccuracy = \frac{S + S_p}{2} \tag{17}$$

5.3 Experimental results and discussion

Dataset was split in ratio 4:1 for training and evaluation. Optimal classifier parameters were identified using random grid search (caret package in R) [k-fold cross-validation, up sampling]. Table 10, and 11 give performance of the classifiers on the train set and Table 12 gives performance of the classifier on the Test set.

Table 10: Evaluating classifier performance on train set

Model	**Train**							
	logloss		**AUC**		**prAUC**		**Accuracy**	
	Mean	**SD**	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**
SVM	2.84	0.01	0.6	0.02	0.18	0.01	0.03	0.005
Logistic Regression	3.03	0.06	0.52	0.04	0.07	0.02	0.09	0.01
XGBOOST	1.68	0.23	0.86	0.01	0.13	0.01	0.86	0.004
Random Forest	1.17	0.13	0.88	0.01	0.15	0.01	0.88	0.002
Proposed Model	0.42	0.02	0.95	0.15	0.26	0.01	0.88	0.01

Table 11: Evaluating classifier performance on train set

Model	**Train**					
	Detection rate		**Mean Specificity**		**Kappa**	
	Mean	**SD**	**Mean**	**SD**	**Mean**	**SD**
SVM	0.0019	0.0002	0.93	0.0007	0.02	0.001
Logistic Regression	0.006	0.0004	0.94	0.01	0.035	0.001
XGBOOST	0.04	0.006	0.92	0.004	0.65	0.03
Random Forest	0.05	0.004	0.99	0.001	0.68	0.01
Proposed Model	0.05	0.003	0.99	0.01	0.68	0.01

Table 12: Evaluating classifier performance on test set

Model	**Test**				
	Accuracy	**95% CI**	**No Information rate**	**P-value**	**Kappa**
SVM	0.0422	0.0204, 0.0762	0.8143	1	0.0245
Logistic Regression	0.0169	0.0046, 0.0426	0.8143	1	0.0026
XGBOOST	0.8916	0.8834, 0.9124	0.7914	0.998e-04	0.7423
Random Forest	0.9241	0.8826, 0.9544	0.8143	1.384e-06	0.7699
Proposed Model	0.91	0.8727, 0.9477	0.8143	9.649e-06	0.7411

The proposed method achieves comparable accuracy to the Random forest with a lower standard deviation. Accuracy achieved on individual classes can be observed from the confusion matrices (see Table 13, 14, 16, 15 and 17). For the confusion matrices, x-axis of the table gives the actual class, and y-axis gives the predicted class of the observation.

Table 13: Confusion matrix of SVM classifier on Test set

	AM	CC	CSP	DM	E	F	G	M	MX	P2P	PG	PZ	P	T	W	UNC
AM	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
CC	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0
CSP	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
DM	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
E	0	0	0	0	6	0	1	0	0	0	0	0	0	0	0	0
F	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0
G	0	0	0	0	7	0	2	0	11	1	0	1	0	0	193	0
M	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
MX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
P2P	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
PG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PZ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P	0	0	0	0	0	0	0	0	1	0	0	2	2	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
UNC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 14: Confusion matrix of Logistic Regression classifier on Test set

	AM	CC	CSP	DM	E	F	G	M	MX	P2P	PG	PZ	P	T	W	UNC
AM	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
CC	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
CSP	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DM	0	0	0	0	0	0	1	0	3	0	0	0	0	0	138	0
E	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
F	0	0	0	1	2	0	0	0	0	0	0	0	0	0	0	0
G	0	0	0	2	9	0	2	0	1	1	0	2	1	0	41	1
M	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P2P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PG	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
PZ	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
P	0	0	0	0	2	0	0	0	8	0	0	0	1	0	14	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
UNC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 15: Confusion matrix of XGBOOST classifier on Test set

	AM	CC	CSP	DM	E	F	G	M	MX	P2P	PG	PZ	P	T	W	UNC
AM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CSP	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DM	0	0	0	3	2	0	1	0	0	0	0	1	0	0	0	0
E	1	0	0	0	7	0	1	0	1	0	0	0	1	0	0	1
F	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
G	0	0	0	0	1	0	2	0	1	0	0	0	0	0	0	0
M	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MX	0	0	0	0	0	0	0	0	10	0	0	0	2	0	0	0
P2P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PG	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
PZ	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	0
P	0	0	0	0	0	0	0	0	1	0	0	2	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W	0	0	0	3	0	0	2	0	0	0	0	0	0	0	187	0
UNC	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Table 16: Confusion matrix of Random Forest classifier on Test set

	AM	CC	CSP	DM	E	F	G	M	MX	P2P	PG	PZ	P	T	W	UNC
AM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CSP	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DM	0	0	0	3	3	0	1	0	0	0	0	0	0	0	0	0
E	0	0	0	0	9	0	1	0	0	0	0	0	1	0	0	1
F	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
G	0	0	0	0	1	0	2	0	1	0	0	0	0	0	0	0
M	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MX	0	0	0	0	0	0	0	0	10	0	0	0	2	0	0	0
P2P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PG	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
PZ	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	0
P	0	0	0	0	0	0	0	0	1	0	0	2	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	193	0
UNC	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Table 17: Confusion matrix of Proposed classifier on Test set

	AM	CC	CSP	DM	E	F	G	M	MX	P2P	PG	PZ	P	T	W	UNC
AM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CC	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
CSP	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DM	0	0	0	2	1	0	1	0	0	0	0	0	0	0	0	0
E	0	0	0	1	11	0	0	0	3	1	0	0	0	0	0	1
F	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
M	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MX	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0
P2P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PG	0	0	0	0	1	0	0	0	0	0	1	0	2	0	0	0
PZ	0	0	0	0	1	0	1	0	0	0	0	1	0	0	0	0
P	0	0	0	0	0	0	0	0	1	0	0	2	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
W	0	0	0	0	0	0	1	0	0	0	0	0	0	0	193	0
UNC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Confusion matrix illustrates that proposed model classified accurately with specificity $0.25-0.73$. Other models with exception of XGBOOST $(0.38-0.94)$ and RF $(0.33-0.83)$ achieved sensitivity $0-0.5$. Table 18, 19, 21, 20 and 22 gives the summary of results for each class obtained by the models.

Table 18: Statistics by Class of SVM classifier on Test set

	AM	CC	CSP	DM	E	F	G	M	MX	P2P	PG	PZ	P	T	W	UNC
Sensitivity	NA	NA	NA	0	0.4	NA	0.5	NA	0	0	0	0	0.5	NA	0	0
Specificity	0.99	0.99	0.99	0.99	0.99	1	0.07	1	0.99	0.99	1	1	0.98	1	1	1
Pos Pred Value	NA	NA	NA	0	0.86	NA	0.009	NA	0	0	NaN	NaN	0.4	NA	NaN	NaN
Neg Pred Value	NA	NA	NA	0.99	0.96	NA	0.9	NA	0.94	0.99	0.99	0.98	0.99	NA	0.18	0.99
Prevalence	0	0	0	0.01	0.06	0	0.016	0	0.05	0.004	0.004	0.012	0.016	0	0.81	0.004
Detection Rate	0	0	0	0	0.02	0	0.008	0	0	0	0	0	0.008	0	0	0
Detection Prevalence	0.004	0.008	0.004	0.004	0.029	0	0.91	0	0.008	0.004	0	0	0.021	0	0	0
Balanced Accuracy	NA	NA	NA	0.49	0.69	NA	0.28	NA	0.49	0.49	0.5	0.5	0.743	NA	0.5	0.5

Table 19: Statistics by Class of Logistic Regression classifier on Test set

	AM	CC	CSP	DM	E	F	G	M	MX	P2P	PG	PZ	P	T	W	UNC
Sensitivity	NA	NA	NA	0	0	NA	0.5	NA	0	0	1	0	0.25	NA	0	0
Specificity	0.99	0.99	1	0.39	0.98	0.98	0.75	1	1	1	1	0.99	0.89	1	1	1
Pos Pred Value	NA	NA	NA	0	0	NA	0.03	NA	NaN	NaN	1	0	0.04	NA	NaN	NaN
Neg Pred Value	NA	NA	NA	0.96	0.93	NA	0.98	NA	0.94	0.99	1	0.98	0.98	NA	0.18	0.99
Prevalence	0	0	0	0.012	0.06	0	0.016	0	0.05	0.004	0.004	0.012	0.016	0	0.81	0.004
Detection Rate	0	0	0	0	0	0	0.008	0	0	0	0.004	0	0.004	0	0	0
Detection Prevalence	0.004	0.004	0	0.59	0.008	0.012	0.253	0	0	0	0.004	0.008	0.105	0	0	0
Balanced Accuracy	NA	NA	NA	0.19	0.49	NA	0.62	NA	0.51	0.5	1	0.495	0.573	NA	0.5	0.5

Table 20: Statistics by Class of XGBOOST classifier on Test set

	AM	CC	CSP	DM	E	F	G	M	MX	P2P	PG	PZ	P	T	W	UNC
Sensitivity	NA	NA	NA	0.89	0.54	NA	0.48	NA	0.76	0	0.89	0.38	0	NA	0.94	0
Specificity	0.91	1	1	0.92	0.88	0.92	0.93	0.94	0.96	1	1	0.99	0.98	1	0.94	0.99
Pos Pred Value	NA	NA	NA	0.46	0.69	NA	0.5	NA	0.93	NaN	1	0.43	0	NA	1	0
Neg Pred Value	NA	NA	NA	0.81	0.89	NA	0.91	NA	0.91	0.94	1	0.94	0.94	NA	1	0.91
Prevalence	0	0	0	0.012	0.08	0	0.018	0	0.06	0.004	0.004	0.012	0.016	0	0.73	0.004
Detection Rate	0	0	0	0.012	0.034	0	0.007	0	0.04	0	0.004	0.004	0	0	0.84	0
Detection Prevalence	0	0	0	0.01	0.05	0.03	0.016	0	0.05	0	0.004	0.012	0.012	0	0.76	0.004
Balanced Accuracy	NA	NA	NA	0.92	0.83	NA	0.78	NA	0.92	0.46	0.95	0.67	0.53	NA	0.86	0.49

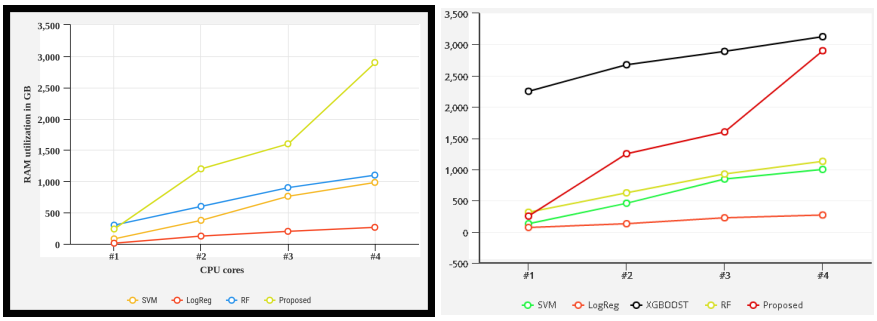
Table 21: Statistics by Class of Random Forest classifier on Test set

	AM	CC	CSP	DM	E	F	G	M	MX	P2P	PG	PZ	P	T	W	UNC
Sensitivity	NA	NA	NA	1	0.6	NA	0.5	NA	0.83	0	1	0.33	0	NA	1	0
Specificity	1	1	1	0.99	0.98	0.99	0.99	1	0.99	1	1	0.99	0.98	1	1	0.99
Pos Pred Value	NA	NA	NA	0.42	0.75	NA	0.5	NA	0.83	NaN	1	0.33	0	NA	1	0
Neg Pred Value	NA	NA	NA	1	0.97	NA	0.99	NA	0.99	0.99	1	0.98	0.99	NA	1	0.99
Prevalence	0	0	0	0.012	0.06	0	0.016	0	0.05	0.004	0.004	0.012	0.016	0	0.81	0.004
Detection Rate	0	0	0	0.012	0.037	0	0.008	0	0.04	0	0.004	0.004	0	0	0.81	0
Detection Prevalence	0	0	0	0.02	0.05	0.04	0.016	0	0.05	0	0.004	0.012	0.012	0	0.81	0.004
Balanced Accuracy	NA	NA	NA	0.99	0.79	NA	0.75	NA	0.91	0.5	1	0.66	0.49	NA	1	0.49

Table 22: Statistics by Class of Proposed classifier on Test set

	AM	CC	CSP	DM	E	F	G	M	MX	P2P	PG	PZ	P	T	W	UNC
Sensitivity	NA	NA	NA	0.66	0.73	NA	0.25	NA	0.66	0	1	0.33	0	NA	1	0
Specificity	1	0.99	1	0.99	0.97	0.99	1	1	0.99	1	0.99	0.98	0.98	1	0.97	1
Pos Pred Value	NA	NA	NA	0.5	0.64	NA	1	NA	0.8	NaN	0.5	0.25	0	NA	0.99	NaN
Neg Pred Value	NA	NA	NA	0.99	0.98	NA	0.98	NA	0.98	0.99	1	0.99	0.98	NA	1	0.99
Prevalence	0	0	0	0.01	0.06	0	0.016	0	0.05	0.004	0.004	0.012	0.016	0	0.81	0.004
Detection Rate	0	0	0	0.008	0.04	0	0.004	0	0.03	0	0.004	0.004	0	0	0.81	0
Detection Prevalence	0	0.004	0	0.016	0.07	0.004	0.004	0	0.04	0	0.008	0.016	0.012	0	0.81	0
Balanced Accuracy	NA	NA	NA	0.83	0.85	NA	0.625	NA	0.83	0.5	0.99	0.66	0.49	NA	0.96	0.5

Resource intensiveness of the proposed model vis-a-vis other ML models was performed (see Figure 4). Four versions of the models (SVM, LogReg, RF, Proposed) using a different number of cores between one and four were built. The utilization of CPU (see Figure 4a) and RAM (see Figure 4b) was monitored.



(a) CPU cores and their utilization (b) RAM utilization

Fig. 4: Efficient utilization of hardware configuration

Figure 4a shows the CPU core utilization $0 - 400$ for unutilized to max utilization of the four CPU cores. SVM and proposed models have high CPU utilization compared to LogReg, XGBOOST and RF. Similarly, RAM utilization for the proposed model is higher compared to the other four models. Hence, the proposed model is resource-intensive as it has higher parameters than the other three models. Higher parameters also result in higher accuracy of predictions. Figure 5 illustrates the precision, recall and f-score (macro) of SVM, LogReg, XGBOOST, RF and the proposed model on the test set. The proposed model achieves results comparable to RF and XGBOOST. The three metrics of the proposed models are $0.9 - 0.94$ due to high classification accuracy for Mixers, exchanges, Ponzi and pools. These four classes dominate the dataset, whereas, classes that were misclassified were affiliatemarketing, criminals, cybersec, trading and miner. These classes represent $< 5\%$ of the dataset, and hence misclassification is not affecting the overall performance of the model.

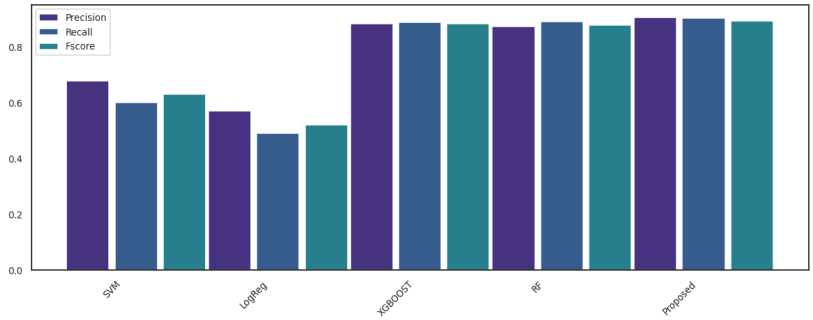


Fig. 5: Precision, Recall and F-score (Macro) on Test set

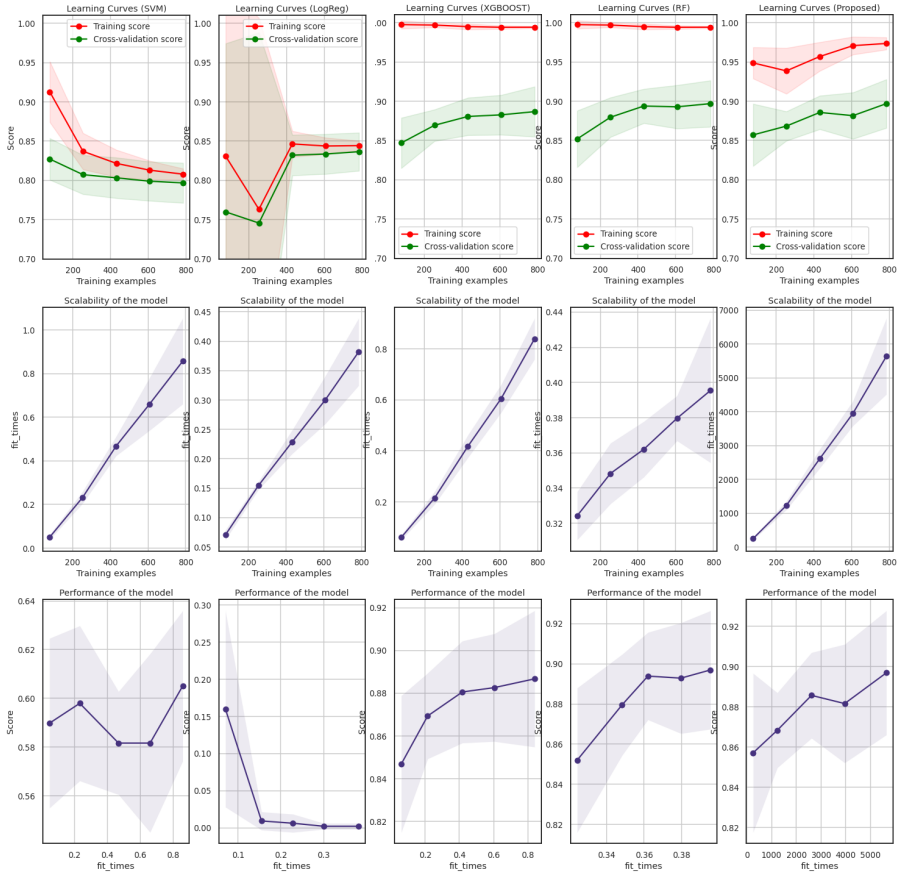


Fig. 6: Training and Cross-validation learning curve, the training samples vs fit times curve, the fit times vs score curve

Figure 6 illustrates the learning curve, scalability and performance of SVM, LogReg, XGBOOST, RF and the proposed models. Learning curves show that by increasing the training examples, the accuracy of the proposed model increases on training and validation set. Learning curves for XGBOOST and RF exhibit overfitting whereas, SVM and LogReg exhibit underfitting. Scalability curves illustrate time taken for training increases exponentially for all models other than RF (linear). Performance of the models indicates the trade-off between fit times and prediction accuracy. It can be observed that lowest fit times are for SVM, and logReg followed by XGBOOST, RF and the proposed model. However, prediction accuracy is highest for proposed model vis-a-vis the rest.

6 Conclusion and Future works

The volume and complexity of Bitcoin Blockchain make machine learning an indispensable tool for forensic investigation. However, issues that machine learning models face in Bitcoin forensics are lack of public, benchmark datasets for training. Lack of ground truth labeled information is seen in Bitcoin, which is not observed in other domains such as image processing and audio processing. Hence, the current work collected a dataset of 1216 Bitcoin users and categorized them into 16 classes. Due to the secrecy provided by Bitcoin, data collection was a mammoth task requiring manual labeling of entities using third-party tools. By allowing open access, the hope is to motivate other researchers by providing a starting point. Nine independent and discriminating features identified for each observation allowed the training of an ensemble decision tree-based model.

Due to higher parameters of the ensemble tree model, the classification accuracy was 0.91, with 95% CI - 0.8727, 0.9477. This was better than two popular models in the literature and comparable to the RF model. The sensitivity and specificity also highlight the efficacy of the proposed method. Compared to existing methods, the number of features used was lower, leading to more accessible feature engineering and model training. Based on the results of the proposed model on the test set, misclassifications of classes of legal businesses to illegal and more seriously illegal businesses to legal was observed. For instance, ponzi as payment gateway or pools, dark markets to exchanges, and exchanges as darkmarkets. This was the results of limited instances of these classes in the training data. This data limitation caused the learning technique to fail at learning discriminative features.

Improving the accuracy of the model could be the next step. This can be possible with more features and with a larger dataset. CPU and RAM utilization could be reduced by feature sampling or regularization techniques. In addition, one-shot or zero-shot learning strategies could be investigated for overcoming issues due to lack of sufficient data.

References

1. Aiolli F, Conti M, Gangwal A, Polato M (2019) Mind your wallet's privacy: Identifying bitcoin wallet apps and user's actions through network traffic analysis. DOI 10.1145/3297280.3297430
2. Akcora CG, Li Y, Gel YR, Kantarcioglu M (2019) Bitcoinheist: Topological data analysis for ransomware detection on the bitcoin blockchain. 1906.07852
3. Alqassem I, Rahwan I, Svetinovic D (2018) The anti-social system properties: Bitcoin network data analysis. IEEE Transactions on Systems, Man, and Cybernetics: Systems

4. Bartoletti M, Pes B, Serusi S (2018) Data mining for detecting bitcoin ponzi schemes. In: 2018 Crypto Valley Conference on Blockchain Technology (CVCBT), pp 75–84
5. Bistarelli S, Mercanti I, Santini F (2018) A suite of tools for the forensic analysis of bitcoin transactions: Preliminary report. In: European Conference on Parallel Processing, Springer, pp 329–341
6. Bogner A (2017) Seeing is understanding: anomaly detection in blockchains with visualized features. In: Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers, pp 5–8
7. Bohannon J (2016) The bitcoin busts
8. Böhme R, Christin N, Edelman B, Moore T (2015) Bitcoin: Economics, technology, and governance. Journal of economic Perspectives 29(2):213–38
9. Bonneau J, Narayanan A, Miller A, Clark J, Kroll JA, Felten EW (2014) Mixcoin: Anonymity for bitcoin with accountable mixes. In: International Conference on Financial Cryptography and Data Security, Springer, pp 486–504
10. Chen T, Guestrin C (2016) Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Association for Computing Machinery, New York, NY, USA, p 785794
11. Conti M, Kumar ES, Lal C, Ruj S (2018) A survey on security and privacy issues of bitcoin. IEEE Communications Surveys & Tutorials 20(4):3416–3452
12. Ermilov D, Panov M, Yanovich Y (2017) Automatic bitcoin address clustering. In: 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), IEEE, pp 461–466
13. Foley S, Karlsen JR, Putniņš TJ (2019) Sex, drugs, and bitcoin: How much illegal activity is financed through cryptocurrencies? The Review of Financial Studies 32(5):1798–1853
14. Gaihre A, Luo Y, Liu H (2018) Do bitcoin users really care about anonymity? an analysis of the bitcoin transaction graph. In: 2018 IEEE International Conference on Big Data (Big Data), IEEE, pp 1198–1207
15. Harlev MA, Sun Yin H, Langenheldt KC, Mukkamala R, Vatrapu R (2018) 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
16. Herrera-Joancomartí J (2014) Research and challenges on bitcoin anonymity. In: Data Privacy Management, Autonomous Spontaneous Security, and Security Assurance, Springer, pp 3–16
17. Hu Y, Seneviratne S, Thilakarathna K, Fukuda K, Seneviratne A (2019) Characterizing and detecting money laundering activities on the bitcoin network. arXiv preprint arXiv:191212060

18. Irwin AS, Turner AB (2018) Illicit bitcoin transactions: challenges in getting to the who, what, when and where. Journal of money laundering control
19. Janda A (2016) Walletexplorer. com: Smart bicoin block explorer
20. Jourdan M, Blandin S, Wynter L, Deshpande P (2018) Characterizing entities in the bitcoin blockchain. In: 2018 IEEE International Conference on Data Mining Workshops (ICDMW), IEEE, pp 55–62
21. Kanemura K, Toyoda K, Ohtsuki T (2019) Identification of darknet markets bitcoin addresses by voting per-address classification results. In: 2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), IEEE, pp 154–158
22. Lee C, Maharjan S, Ko K, Hong JWK (2020) Toward detecting illegal transactions on bitcoin using machine-learning methods. In: Zheng Z, Dai HN, Tang M, Chen X (eds) Blockchain and Trustworthy Systems, Springer Singapore, Singapore, pp 520–533
23. Liang J, Li L, Luan S, Gan L, Zeng D (2019) Bitcoin exchange addresses identification and its application in online drug trading regulation
24. Liu T, Ge J, Wu Y, Dai B, Li L, Yao Z, Wen J, Shi H (2020) A new bitcoin address association method using a two-level learner model. In: Wen S, Zomaya A, Yang LT (eds) Algorithms and Architectures for Parallel Processing, Springer International Publishing, Cham, pp 349–364
25. Maesa DDF, Marino A, Ricci L (2016) Uncovering the bitcoin blockchain: an analysis of the full users graph. In: 2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA), IEEE, pp 537–546
26. Maesa DDF, Marino A, Ricci L (2018) Data-driven analysis of bitcoin properties: exploiting the users graph. International Journal of Data Science and Analytics 6(1):63–80
27. Maesa DDF, Marino A, Ricci L (2018) The graph structure of bitcoin. In: International Conference on Complex Networks and their Applications, Springer, pp 547–558
28. Maesa DDF, Marino A, Ricci L (2019) The bow tie structure of the bitcoin users graph. Applied Network Science 4(1):56
29. Monamo P, Marivate V, Twala B (2016) Unsupervised learning for robust bitcoin fraud detection. In: 2016 Information Security for South Africa (ISSA), IEEE, pp 129–134
30. Monamo PM, Marivate V, Twala B (2016) A multifaceted approach to bitcoin fraud detection: Global and local outliers. In: 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA), IEEE, pp 188–194
31. Nakamoto S (2019) Bitcoin: A peer-to-peer electronic cash system. Tech. rep., Manubot
32. Nan L, Tao D (2018) Bitcoin mixing detection using deep autoencoder. In: 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC), pp 280–287
33. Paquet-Clouston M, Romiti M, Haslhofer B, Charvat T (2019) Spams meet cryptocurrencies: Sextortion in the bitcoin ecosystem. In: Proceedings of

the 1st ACM Conference on Advances in Financial Technologies, pp 76–88
34. Park S, Im S, Seol Y, Paek J (2019) Nodes in the bitcoin network: comparative measurement study and survey. IEEE Access 7:57009–57022
35. Pham T, Lee S (2016) Anomaly detection in bitcoin network using unsupervised learning methods. arXiv preprint arXiv:161103941
36. Phetsouvanh S, Oggier F, Datta A (2018) Egret: Extortion graph exploration techniques in the bitcoin network. In: 2018 IEEE International Conference on Data Mining Workshops (ICDMW), pp 244–251
37. Pinna A, Tonelli R, Orrú M, Marchesi M (2018) A petri nets model for blockchain analysis. The Computer Journal 61(9):1374–1388
38. Portnoff RS, Huang DY, Doerfler P, Afroz S, McCoy D (2017) Backpage and bitcoin: Uncovering human traffickers. In: KDD '17
39. Rahouti M, Xiong K, Ghani N (2018) Bitcoin concepts, threats, and machine-learning security solutions. IEEE Access 6:67189–67205
40. Reyes-Macedo VG, Salinas-Rosales M, Garcia GG (2019) A method for blockchain transactions analysis. IEEE Latin America Transactions 17(07):1080–1087
41. Sabry F, Labda W, Erbad A, Al Jawaheri H, Malluhi Q (2019) Anonymity and privacy in bitcoin escrow trades. In: Proceedings of the 18th ACM Workshop on Privacy in the Electronic Society, pp 211–220
42. Shao W, Li H, Chen M, Jia C, Liu C, Wang Z (2018) Identifying bitcoin users using deep neural network. In: Vaidya J, Li J (eds) Algorithms and Architectures for Parallel Processing, Springer International Publishing, Cham, pp 178–192
43. Sun Yin HH, Langenheldt K, Harlev M, Mukkamala RR, Vatrapu R (2019) Regulating cryptocurrencies: a supervised machine learning approach to de-anonymizing the bitcoin blockchain. Journal of Management Information Systems 36(1):37–73
44. Toyoda K, Mathiopoulos PT, Ohtsuki T (2019) A novel methodology for hyip operators bitcoin addresses identification. IEEE Access 7:74835–74848
45. Turner A, Irwin ASM (2018) Bitcoin transactions: a digital discovery of illicit activity on the blockchain. Journal of Financial Crime
46. Vasek M, Moore T (2015) There's no free lunch, even using bitcoin: Tracking the popularity and profits of virtual currency scams. In: Böhme R, Okamoto T (eds) Financial Cryptography and Data Security, Springer Berlin Heidelberg, Berlin, Heidelberg, pp 44–61
47. Weber M, Domeniconi G, Chen J, Weidele DKI, Bellei C, Robinson T, Leiserson CE (2019) Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics. arXiv preprint arXiv:190802591
48. Wu Y, Luo A, Xu D (2019) Identifying suspicious addresses in bitcoin thefts. Digital Investigation 31:200895, DOI 10.1016/j.fsidi.2019.200895
49. Wu Y, Tao F, Liu L, Gu J, Panneerselvam J, Zhu R, Shahzad MN (2020) A bitcoin transaction network analytic method for future blockchain forensic investigation. IEEE Transactions on Network Science and Engineering pp

1–1
50. Yang L, Dong X, Xing S, Zheng J, Gu X, Song X (2019) An abnormal transaction detection mechanim on bitcoin. In: 2019 International Conference on Networking and Network Applications (NaNA), IEEE, pp 452–457
51. Yin HS, Vatrapu R (2017) A first estimation of the proportion of cybercriminal entities in the bitcoin ecosystem using supervised machine learning. In: 2017 IEEE International Conference on Big Data (Big Data), IEEE, pp 3690–3699
52. Zarpelão BB, Miani RS, Rajarajan M (2019) Detection of bitcoin-based botnets using a one-class classifier. In: Blazy O, Yeun CY (eds) Information Security Theory and Practice, Springer International Publishing, Cham, pp 174–189
53. Zayuelas Muñoz J (2019) Detection of bitcoin miners from network measurements. B.S. thesis, Universitat Politècnica de Catalunya
54. Zhang Z, Zhou T, Xie Z (2017) Bitscope: Scaling bitcoin address de-anonymization using multi-resolution clustering
55. Zola F, Eguimendia M, Bruse JL, Urrutia RO (2019) Cascading machine learning to attack bitcoin anonymity. In: 2019 IEEE International Conference on Blockchain (Blockchain), IEEE, pp 10–17