Illegal Activity Detection on Bitcoin Transaction using Deep Learning

Pranav ajeet nerurkar (pranav.n@nmims.edu)

Narsee Monjee Institute of Management Studies University https://orcid.org/0000-0002-9100-6437

Research Article

Keywords: Cryptocurrency, Cybercrime, Deep learning, Neural networks

Posted Date: September 15th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1454891/v1

Illegal Activity Detection on Bitcoin Transaction using Deep Learning

Pranav Nerurkar

Received: May 2020

Abstract Forensic investigations increasingly leverage artificial intelligence to identify illegal activities on Bitcoin. Bitcoin transactions have an original graph (network) structure, which is sophisticated and yet informative. However, machine learning applications on Bitcoin have given limited attention to developing end to end deep learning frameworks that are modeled to exploit the Bitcoin graph structure. To identify illegal transactions on Bitcoin, the current paper extracts nineteen features from the bitcoin network and proposes a deep learning based graph neural network model using spectral graph convolutions and transaction features. The proposed model is compared with two state of the art techniques an Ensemble of Decision Trees, and a Decision trees trained on Convoluted features for classification of illegal transactions on Bitcoin. To understand the efficacy of the proposed model, a dataset is collected consisting of 13310125 transactions of 2059 entities having 3152202 Bitcoin account addresses and belonging to 28 categories of users. Two sets of experiments are performed on the datasets: labeling transactions as legal or illegal (binary classification) and identifying the originator of the transaction to one of the twenty-eight types of entities (multi-class classification). For fast and accurate decisions, binary classification is appropriate, and for pinpointing the category of bitcoin users, a multi-class classifier is suitable. On both the tasks, the proposed models achieved a maximum of 92% accuracy, validating the methodology and suitability of the model for real-world deployment.

Keywords Cryptocurrency Cybercrime Deep learning Neural networks

Pranav Nerurkar
Dupuy de Lome Research Institute of Universite Bretagne, Sud, France
Dept. of Data Science, MPSTME, NMIMS University, Mumbai, India
E-mail: pranav-ajeet.nerurkar@univ-ubs.fr, pranav.n@nmims.edu

1 Background

Since its initiation in 2009, Bitcoin [1] has been buried in contentions for giving a sanctuary to criminal operations. A few sorts of unlawful clients take cover of secrecy or anonymity, and revealing such kind of elements is crucial for forensic or cybercrime investigations [1–5]. Fervent pundits of Bitcoin guarantee that it is anti-social and anti-transparency as it makes obstacles for law enforcement to follow dubious exchanges because of the anonymity and security [6–8]. Since Bitcoin's outstanding growth in transactions during 2012-2016, clients viz. mixing services [9], betting destinations, exchanging trades, financial specialists, examiners, and autonomous mining enterprises [10] have entered the Bitcoin biological system. The 2012-onwards stage saw the development of Ponzi plans, illegal tax avoidance, cheats [11], misappropriations, blackmail [12, 13] and tax avoidance [14] strategies that utilized the cover of anonymity afforded by Bitcoin cryptocurrency to misdirect the review trail. It was theorized that in 2017, BTCs of the value $770 million were exchanged for unlawful exercises [15], a fourth of bitcoin clients were noxious and 46% of all bitcoin action was illicit [16].

To stay up with the illegal action, legal apparatuses need to investigate voluminous information created by bitcoin exchanges on the Blockchain. A panacea was offered by AI which turned into a mainstream procedure for following and investigating unlawful clients or exchanges. Existing writing studied on distinguishing criminal operations utilizing Machine Learning (ML) had zeroed in on deanonymizing elements [17–20], recognizing botnets [21], unlawful exchanges [15], distinguishing dubious bitcoin clients [4, 5, 22–27] (extortionists [28], ponzi tricks [29], darknet markets [30], ransomwares [31], human dealers [32], frauds [33, 34]), recognizing tax evasion [10, 35, 36], distinguishing blending administrations [37], recognizing bitcoin trades [38], distinguishing illicit exchanges [39, 40], distinguishing bitcoin payment services [41] and bitcoin mining companies [42]. The methodology defined for such examinations is given in Figure 1.

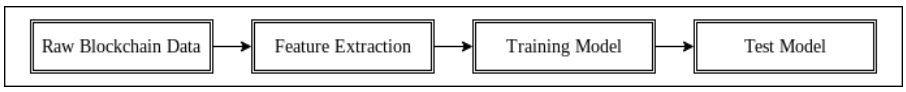

Fig. 1: Steps of ML on Bitcoin system

Feature engineering is basic for ML applications, and the degree for feature engineering and extraction in Bitcoin is huge because of the general classifications of metadata related with Blockchain (see Figure **??**). AI or deep learning has achieved outlook changes in displaying entities in domains, for example, image identification, object localization, or signal or speech processing. Be that as it may, AI has gained limited progress in domains like cryptocurrencies because of absence of benchmark, public datasets (see Table 3) [19], absence of preparing ability or apparatuses to deal with full data of Blockchain, and absence of ground truth data on the characters

[1] Bitcoin alludes to the system, and bitcoin or BTC alludes to the digital currency

of bitcoin clients. Aside from these issues, pseudo-obscurity in digital currencies permits clients to execute with one another through hash address or keys. These keys are reusable and also can be created and discarded infinitely, this makes the techniques for assimilating transactions to a user.

1.1 Motivation

Existing studies predominantly focused on feature engineering/extraction followed by supervised learning (see Table 2) to identify illicit activities on Bitcoin. Such investigations require deanonymizing methods (see Section 2.3) to connect numerous keys to a solitary entity. Indeed, even the most mainstream deanonymizing procedures have limits because of the utilization of mixing services [43, 44]. Besides, to lessen the computational multifaceted nature of AI models, the objective of interest was confined to restricted classes of illegal clients. Also, the time stretch for which information was gathered from the Blockchain for include designing was confined to more limited ranges. Due to such factors, the techniques that were to be used for inference generation after training protocols were not generalized. The issues mentioned above tend to limit the efficacy of forensic investigations. Deviating from existing methodologies for detecting illegal activities on Bitcoin, the current paper proposes supervised learning approaches on the Bitcoin transactions' network structure.

1.2 Contributions

- This paper represents the largest study in our knowledge conducted on the identification of suspicious Bitcoin addresses - the dataset collected for use in this paper contains 13310125 transactions of 2059 entities having 3152202 Bitcoin addresses and belonging to 28 categories
- Released the dataset and script to motivate further research
- Feature engineering to identify the optimal transaction patterns observed in illegal activities.
- Proposed models for supervised learning for detecting illegal activities on Bitcoin
- Conducted extensive experiments on transaction graphs from 2009-2020 to validate the proposed approach and highlighted future works

1.3 Novelty

The current research is of significant importance to the development of Bitcoin forensic investigation tools. The methodology relies on deep learning on the transaction graph of the Bitcoin network. Handcrafted feature engineering is minimized, and a data-driven approach to learning features from the transaction graph is used. An additional benefit of the proposed approach is that it processes the transaction graph and avoids the preprocessing step of deanonymizing addresses. Although the transaction graph of Bitcoin has been used in previous studies to track the flow of funds [45] or in case

studies investigating individual scams [12, 46, 47], a large scale modeling of illegal activities using transaction graph was uncharted.

1.4 Outline

The remaining parts of the paper are divided as follows: Preliminaries (see Section 2.1, 2.2 and 2.3) and State of the art research (see Section 2.4) in Bitcoin forensics are described in Section 2. Dataset collection and preprocessing along with the methodology are outlined in Section 3. Section 4 gives the mathematical model for the proposed semi-supervised learning for detecting illegal activities. Experimental study and discussion are in Section 5 followed by lessons learnt and the next studies in Section 6.

2 Bibliographic studies

Data structure and reference implementation of Bitcoin and principal ideas, for example, blocks, Blockchain, exchanges, input keys, output keys, service providers on Bitcoin, deanonymization are depicted in Sections 2.1, 2.2, and 2.3. Followed by basic investigation of published research on distinguishing illicit clients (see Section 2.4), research gaps in public databases (see Section 2.5) and machine learning models utilized in bibliographic studies (see Section 2.6).

2.1 Data structure and reference implementation of Bitcoin

Bitcoin exchanges are appended to ”Blocks" and recorded after consensus into a distributed public record ”Blockchain." Each exchange has a few data sources (senders) and outputs (recipients). The metadata [2] related with blocks, exchanges, data sources and outputs gives scope for further examination. A solitary bitcoin client can produce various addresses for sending and accepting BTCs, which makes an impediment in investigating bitcoin clients. Deanonymizing procedures give an answer for overcoming this issue.

2.2 Common kinds of users on Bitcoin

- Exchanges (E): Permit exchanging of BTC to fiat monetary standards
- Pools (P): Individual clients join their preparing power for mining blocks
- Gambling (G): Permit putting down of wagers utilizing BTCs
- Wallets (W): Storage BTC private keys and equilibrium
- Payment gateways (PG): Permit accepting payment for services in BTCs
- Miner (M): Organizations contending to mine blocks
- Darknet markets (DM): Selling and purchasing products utilizing BTCs

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

- Mixers (MX): Remove discernibility of BTCs from source
- Trading locales (T): Purchase securities utilizing BTCs
- P2Plenders (P2P): Crowdsourcing BTCs for credits
- Faucets (F): Reward in BTCs to endorsers
- Explorer (EX): Educational sites give API to investigate Bitcoin
- P2PMarket (P2PM): Marketplace for recycled merchandise where purchasers can contact merchants and make payments in BTCs
- Bond markets (B): Buying securities or financial instruments in BTC
- Affiliate advertisers (AM): Pay per click in BTC
- Video sharing (VM): Payment in BTCs for review recordings
- Money launderers (ML): Convert fiat monetary standards to BTC
- Cyber-security suppliers (CSP): Provide network protection items for BTC
- Cyber-lawbreakers (CC): Blacklisted by governments
- Ponzi (PZ): High yield speculation scams

2.3 Bitcoin cryptocurrency and Deanonymization

Block is a set of exchanges $T = \{t_1, t_2, ..., t_n\}$. For each $t_i \subset T$ there is a 3-tuple (t_s, I^{t_i}, O^{t_i}) where t_s signifies UNIX timestamp of t_i and I, O means the locations of data sources (senders) and outputs (beneficiaries) in t_i separately [21]. Each t_i can have a few data sources and yields i.e., $I^{t_i} = \{i_1, i_2, ..., i_n\}$ and $O^{t_i} = \{o_1, o_2, ..., o_n\}$. Each bitcoin client $u_i \subset U$ where $U = \{u_1, u_2, u_3, ..., u_n\}$ can have numerous addresses and play out various transactions. For sending bitcoins (BTCs), u_i can produce another address for every exchange t_i.

The undertaking of a deanonymizing function $f(.)$ is consolidating all addresses produced 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 exchanges. Here ${i_{t_1}}^{u_i}$ is address created by u_i to send BTCs in t_1 and ${o_{t_n}}^{u_i}$ is address produced by u_i to get BTCs in t_n.

Deanonymizing is a non-trivial methodology because of the multifaceted nature and variety of the Bitcoin network [43, 44]. Functions described in the literature can be arranged as heuristic-based [7, 14, 21, 45, 47], disseminated network-based [48] and AI based [17]. Heuristic-based capacities that are well known and utilized in Bitcoin research papers.

2.4 Bibliographic studies on detecting illegal activities in Bitcoin cryptocurrency

Existing methodologies on Bitcoin forensics can be grouped into three categories (see Figure 2), with AI-based techniques being most popular.

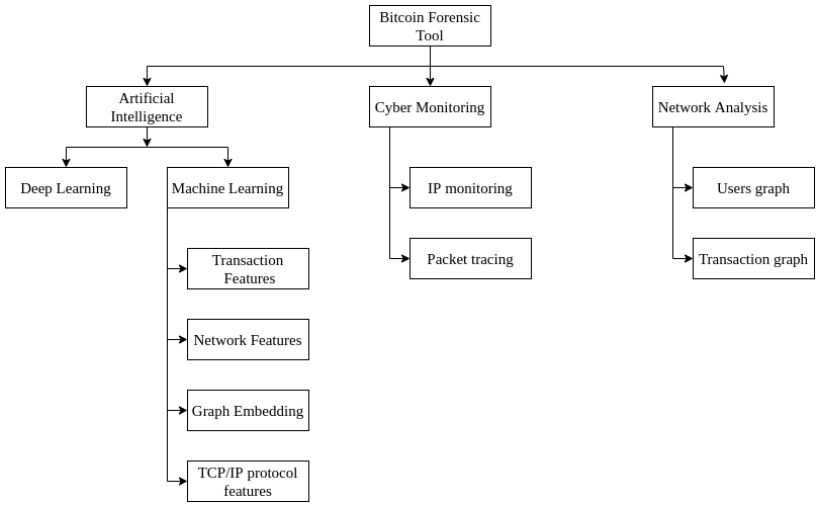

Fig. 2: Categories of Bitcoin forensic studies

A benefit in the investigation of digital cryptocurrencies forms of money is that transaction exchange records are kept up on an distributed record ”Blockchain", which is transparently accessible for assessment. The volume of the Blockchain presents issues in examining it, restricting the interval of time of study, or confining the destinations were utilized by concentrates in the writing to conquer this issue.

Table 1: Summary of distributed bitcoin studies

Authors	**Description**	**Features extracted**
B Zarpelao *et al.* [21]	Detection of botnets utilizing bitcoin protocols to dispatch DDoS attacks	Transaction features
T Liu *et al.* [17]	Deanonymize bitcoin address	Network based features
C Lee *et al.* [15]	Detecting Illegal Transactions on Bitcoin	Transaction features
Y Wu *et al.* [22, 23]	Tracing dubious bitcoin elements	Transaction features
M Weber *et al.* [24]	Identifying illegal bitcoin clients	Transaction features
Y Hu *et al.* [10, 35, 36]	Detecting Money Laundering Activities	Graph embeddings
H Yin *et al.* [25]	Identifying unlawful bitcoin clients	Transaction features
L Nan *et al.* [37]	Mixing administration recognition	Graph embeddings
L Yang *et al.* [26]	Identifying illegal bitcoin clients	Transaction features
J Liang *et al.* [38]	Bitcoin Exchange Identification	Graph embeddings
Z Zhang *et al.* [27]	Identifying illegal bitcoin clients	Transaction features
T Pham *et al.* [39]	Detecting Illegal Transactions on Bitcoin	Clustering hubs on exchange features
features A Bogner [40]	Detecting Illegal Transactions on Bitcoin	Clustering hubs on exchange features
F Zola *et al.* [18]	Deanonymize bitcoin address	Transaction features
F Aiolli *et al.* [41]	Identifying bitcoin wallets	Transaction features
W Shao *et al.* [19]	Deanonymize bitcoin address	Transaction features
M Vasek *et al.* [49]	Identifying bitcoin tricks	Transaction and organization features
M Bartoletti *et al.* [29]	Identifying bitcoin ponzi plans	Transaction and organization features
P Monamo *et al.* [33, 34]	Identifying bitcoin extortion plans	Clustering hubs on exchange features
J Munoz [42]	Identifying bitcoin diggers	Network traffic features
An Irwin *et al.* [4, 5]	Identifying illegal bitcoin clients	Transaction features
R Portnoff *et al.* [32]	Identifying human dealers in bitcoin	Transaction features
C Ackora *et al.* [31]	Identifying ransomware in bitcoin	Transaction features
K Kanemura *et al.* [30]	Identifying darknet advertises in bitcoin	Transaction features
M Jordan *et al.* [20]	Deanonymize bitcoin address	Transaction features
S Phetsouvanh *et al.* [28]	Identifying criminals in bitcoin	Transaction and organization features

Writing on recognizing criminal operations has zeroed in on deanonymizing elements [17–20], distinguishing botnets [21], illicit exchanges [15], distinguishing dubious bitcoin clients [4, 5, 22–27] (scoundrels [28], ponzi tricks [29], darknet markets [30], ransomwares [31], human dealers [32], fakes [33, 34]), distinguish tax evasion [10, 35, 36], distinguishing blending administrations [37], recognize bitcoin trades [38], recognize unlawful exchanges [39, 40], distinguishing bitcoin wallets [41] and bitcoin excavators [42]. Table 1 sums up the systems utilized in these investigations.

Feature extraction is the most basic part of bitcoin examinations centering with respect to unlawful movement or illegal client recognition. Different methodologies utilized by the creators for highlight designing can be gathered into five kinds (see Table 2).

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 Databases utilized in published bibliographic 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 [2, 50–54]) have focused on limited categories of illicit users and shorter periods.

Table 3: Datasets used in published bitcoin studies

Dataset	Closed-access	Features	Categories	Size
Chainanalysis [25, 35, 36]	Yes	9	exchange, gambling, hosted wallet, merchant services, miningpool, mixing, ransomware, scam, tor market or other	198,097,356
Univ. Illinois Urbana-Champaign [33, 34, 39]	No	0	0	37,450,461
BitcoinPonzi [29]	No	11	Ponzi, Non-ponzi	6432
R Portnoff *et al.* [32]	Yes	2	Sex offender, Ordinary	753,929
D Ermilov *et al.* [48]	Yes	238	Service, gambling, mixer, exchange, pool, darknet	244,030,115
Ellipse [24]	No	6	licit, illicit	203,769
C Lee *et al.* [15]	Yes	2	licit, illicit	2 million
M Vasek *et al.* [49]	Yes	2	Ponzi schemes, mining scams, scam wallets and fraudulent exchange	192
Wei Shao *et al.* [19]	Yes	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**	**Accuracy**
k-Nearest Neighbours	[36]	< 0.7
Random Forests (RF)	[36]	0.73
Extra Trees	[36]	< 0.7
Decision Trees	[36]	< 0.7
Bagging Classifier	[36]	0.74
Gradient Boosting	[36]	0.77-0.78
AdaBoost	[36]	0.78
Support Vector Machine (SVM)	[35]	0.76
MultiLayer Perceptron	[35]	0.76
K-Means	[39]	< 0.7
Graph convolutional network	[24]	< 0.7
Logistic regression (LogReg)	[38]	0.85
DeepWalk, Node2vec, SDNE	[10]	0.71-0.78

From Table 1 in writing, it is obvious that the execution of a solid and secure illicit client location framework is a significant worry for protection and security in Bitcoin. Existing works have not zeroed in on an expansive range of criminal operations that are directed on Bitcoin. Furthermore, existing datasets to are inadmissible for AI as their source is restricted. In this regard, in Section 3, depicts the information assortment technique to conquer the issue of information accessibility in open datasets. Section 4 examines the proposed classifier that could distinguish a wide range of unlawful clients on Bitcoin.

3 Materials and Methods

Procedure for Bitcoin dataset collection and preprocessing (see Section 3.1) and construction of transaction graph from Bitcoin data are described.

3.1 Database cleaning

Bitcoin database was constructed using public bigquery repository [3]. All blocks and transactions from 03 Jan 2009 12:45:05 GMT to 08 May 2020 13:21:33 GMT were present in the database. The transaction graph was constructed from transactions occurring within a calendar year. The transaction graph is a tuple $G = (V, E)$ where V is a (finite) set of vertices, and E is a finite collection of edges. The set E contains elements from the union of the one and two-element subsets of V. In the Bitcoin transaction graph G, the V is transaction hashes. E represents the interactions between the users through the exchange of bitcoins. The amount of BTC transferred is an attribute of E. Table 5 gives the notations used to graphically illustrate a typical transaction graph G (see Figure 3) that depicts the flow of BTC's between three typical Bitcoin users viz. Alice, Bob, and Carol.

[3] https://github.com/blockchain-etl/bitcoin-etl

Table 5: Table of notations

Notation	Description
a, b, c	Bitcoin accounts of alice, bob, carol respectively
T = (I, O)	Transaction of Bitcoins; involves BTC transfer between users
I	Input set of a Transaction (T); it may contain one or more credited accounts of users
O	Output set of a Transaction (T); it may contain one or more accounts of users to be credited
$a_{i1}, ..., a_{im}, m \geq 1$	Alice's credited accounts
$b_{i1}, ..., b_{im}, m \geq 1$	Bob's credited accounts
$c_{i1}, ..., c_{im}, m \geq 1$	Carol's credited accounts
v(i)	value in BTC present in an account

Multiple accounts belonging to a single user need to be identified, for this purpose, multi-input heuristic clustering [50–53] was used utilizing an API [4] [55]. This helped in synthesizing a ground truth labelled database for semi-supervised learning (see Table 6) having 3310125 transactions of 2059 entities having 31522025 accounts belonging to 28 categories. Transaction graphs were directed, acyclic, and consisted of valid (defined in [56]) and coinbase transactions. Description of users in Table 6 is given in Section 2.2, "unclassified" refers to users not falling in other 27 categories.

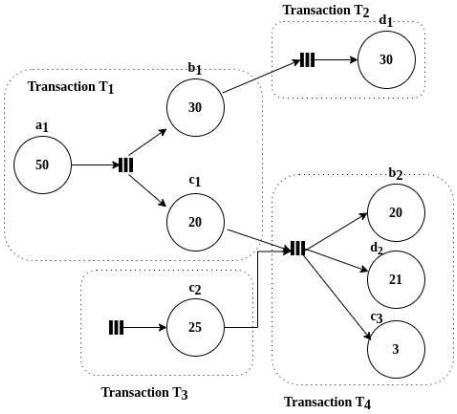

Fig. 3: Modeling transaction graph (G) from Bitcoin

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

Table 6: Types of Bitcoin users in dataset

affiliatemarketing	**blackmail**	**bomb**	**bond**
2	53	1	1
criminals	**cybersec**	**darkmarket**	**donations**
1	2	16	48
exchange	**explorer**	**faucet**	**gambling**
88	2	2	35
laundering	**microworker**	**miner**	**mixer**
1	1	3	53
p2plender	**p2pmarket**	**paymentgateway**	**ponzi**
6	1	6	28
pools	**ransomwares**	**scams**	**sextortionist**
8	11	23	57
trading	**Unclassified**	**videosharing**	**wallets**
9	1592	1	8

The count of transactions per category is in Table 7. 35% of the transactions are created by illegal entities viz. dark market, blacklist, gambling, criminals, mixer, Ponzi, ransom, sextortionists, laundering, scams, and bomb threats.

Table 7: Count of Transactions in each category in Bitcoin transaction dataset

label	**Count**	**label**	**Count**	**label**	**Count**
darkmarket	1045185	donations	728841	bond	48842
pools	458472	gambling	738823	explorer	126592
exchange	2504996	criminals	499760	cybersec	1586
blacklist	1288040	p2pmarket	101452	ransom	7566
trading	236031	mixer	826971	sextort	1794
paymentgateway	1317065	-	-	affiliatemarketing	3900
wallets	1744511	unclassified	227230	laundering	2217
p2plender	595768	videosharing	3593	microworker	4722
faucet	369860	ponzi	284653	scams	187
		miner	141468	bomb	1

Table 8: Distribution of Transactions in each category in Bitcoin transaction dataset

label	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020
darkmarket	-	-	-	-	-	134506	534151	318001	11848	5899	2584	2
pools	-	-	1130	10766	31208	43847	44888	312443	77680	52909	31820	993
exchange	-	-	2796	16530	115752	772656	782175	574958	660612	426323	196292	7278
blacklist	-	-	-	1	53	955	5198	12554	357738	252016	131518	396094
trading	-	-	-	-	8375	24983	4620	2585	85221	122015	49379	1067
paymentgateway	-	-	761	13017	-	620370	152086	406547	38675	-	6343	49
wallets	-	-	29	573	363	20252	329934	964229	509576	458950	257453	12256
p2plender	-	-	-	2514	-	131203	339505	163218	63941	-	30588	32
faucet	-	-	-	93	131	1643	87271	162855	82171	94585	66676	5152
donations	-	8613	84425	93993	-	43481	105739	35452	43055	-	135932	225964
gambling	-	-	-	563	19444	137009	242031	212269	171919	47803	14394	977
criminals	-	-	-	-	-	-	-	-	117468	-	262583	43873
p2pmarket	-	-	-	-	-	9310	33968	61540	4551	2963	1879	9
mixer	-	-	3298	27046	75335	190360	159117	309717	120037	34600	24986	3825
unclassified	53	17801	10689	18591	-	19286	22888	29848	36553	-	76516	12190
videosharing	-	-	-	-	-	356	2072	1961	381	1235	95	-
ponzi	-	-	-	-	124	45298	7433	4046	34600	85813	109379	35439
miner	-	-	-	-	-	4333	13182	11913	10676	8037	86298	40560
bond	-	-	-	-	140	2020	6337	15048	18673	14037	7278	234
explorer	-	-	-	-	-	86	39344	92885	13754	10580	3573	-
cybersec	-	-	-	-	-	620	1101	138	138	127	121	-
ransom	-	-	-	-	344	1818	2290	2196	1086	160	24	2
sextort	-	-	-	-	-	-	-	-	-	1788	187	-
affiliatemarketing	-	-	-	-	-	-	2068	1917	453	184	53	-
laundering	-	-	-	-	-	1078	1151	70	8	6	2	-
microworker	-	-	-	-	-	559	2441	1722	640	192	10	-
scams	-	-	-	-	-	-	-	-	-	-	184	3
bomb	-	-	-	-	-	-	-	-	-	1	-	-

Table 9 gives the features extracted from Bitcoin blockchain for each transaction (tx). Description of the features is listed:

- txsize: The size of this transaction (tx) in bytes
- txvirtualsize: The virtual transaction size (differs from size for SegWit tx)
- txinputs_count: total inputs to a tx
- txoutputs_count: total outputs to a tx
- txinput_val: total BTCs sent by inputs in a tx
- txoutput_val: total BTCs sent to outputs in a tx
- txfee: The fee paid by this transaction
- Min_received: minimum BTCs received in a tx by outputs
- Max_received: maximum BTCs received in a tx by outputs
- Avg_received: average BTCs received in a tx by outputs
- Total_received: total BTCs received in a tx by outputs
- Stdev_received: standard deviation of BTCs received in a tx by outputs
- Var_received: variance of BTCs received in a tx by outputs
- Min_sent: minimum BTCs sent in a tx by inputs
- Max_sent: maximum BTCs sent in a tx by inputs
- Avg_sent: average BTCs sent in a tx by inputs
- Total_sent: total BTCs sent in a tx by inputs
- Stdev_sent: standard deviation of BTCs sent in a tx by inputs
- Var_sent: variance of BTCs sent in a tx by inputs

Table 9: Feature vector for each Transaction in each category in Bitcoin transaction dataset

txsize	txvirtualsize	txinputs_count	txoutputs_count	txinput_val
txoutput_val	txfee	Min_received	Max_received	Avg_received
Total_received	Stdev_received	Var_received	Min_sent	Max_sent
Avg_sent	Total_sent	Stdev_sent	Var_sent	

Figure 4 illustrates the methodology adopted to test the efficacy of the proposed model (see Section 4) using the Bitcoin dataset.

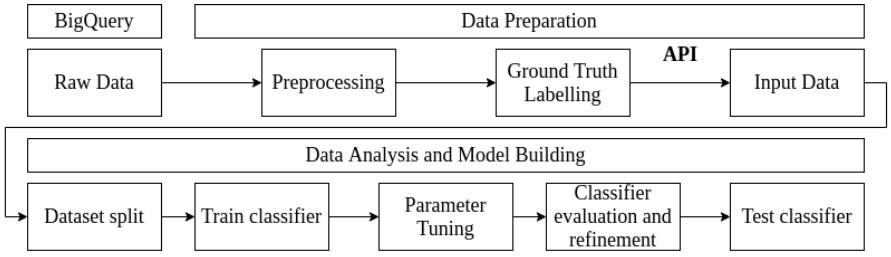

Fig. 4: Flowchart of proposed work

Table 10 gives the graph measurements - order and size of each graph in the Bitcoin transaction dataset.

Table 10: Description of networks in Bitcoin transaction dataset

	Node count	**Edge count**
2009	53	14
2010	25585	60476
2011	96498	267091
2012	152427	460445
2013	673482	2195180
2014	2004286	11023329
2015	2460808	14527531
2016	3256614	14806657
2017	2008800	8656427
2018	1536087	6538288
2019	1124493	5417366
2020	754533	2779216

3.2 Experimental setup

The analyses were completed on a solitary center 1 TB Intel(R) Xeon(R) Silver 4114 CPU@2.20GHz. Programming interface calls were made through the Curl bundle of

R utilizing Jupyter note pads facilitated on Google Colab (n1-highmem-2 example, 2vCPU @2.20GHz, 13GB RAM) and Kaggle (Intel(R) Xeon(R) CPU @2.20GHz, 13GB RAM).

4 Illegal Activity Detection on Bitcoin Transaction using Deep Learning

Mathematical models for the proposed supervised learning approach viz., deep learning based graph neural network model using spectral graph convolutions (see Section 4.1) is given (denoted as Model IIIA and IIIB). Two state of the art approaches are also described further, viz., Ensemble of Decision Trees (Model-II) and vanilla graph convolutional network features (Model-I) for detecting illegal activities in Bitcoin transaction graph with steps followed to train them on the dataset. Figure 5 illustrates the difference between the proposed approach and others (A=adjacency matrix of the Bitcoin transaction dataset, X=feature matrix of transactions).

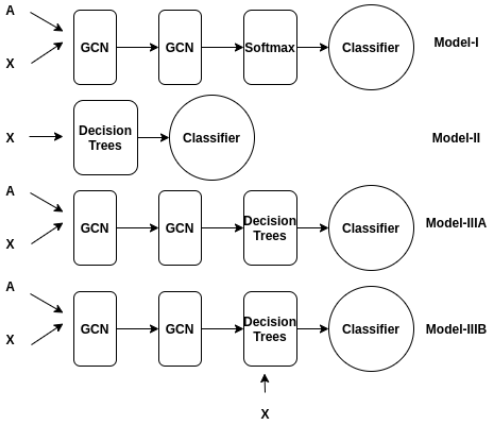

Fig. 5: Difference between the proposed supervised learning model and others

4.1 GCN based classifier (Model-IIIA and IIIB)

The architecture of model contains two stacked Graph Convolutional Network (GCN) Layers followed by a single fully connected layer. A Bitcoin transaction graph $G(V, E)$ with V as the node set of transactions $T_i, T_j, ..., T_n$; $m \geq 1$ and E as the dyad set representing BTC transfers between accounts has a binary adjacency matrix $A \in \mathbb{R}^{|V|*|V|}$ and a node feature matrix $X \in \mathbb{R}^{|V|*m}$. The node feature matrix is formed by stacking horizontally the m-dimension feature vector of each node (see Table 9). GCN has the following steps:

1. Apply the convolutional filter $\hat{A}$ is constructed from the new adjacency matrix $\tilde{A} = A + I$ and new degree matrix $\tilde{D} = D + I$ by the operation $\hat{A} = \tilde{D}^{\frac{-1}{2}} \tilde{A} \tilde{D}^{\frac{-1}{2}}$

2. Use the propagation rule (see Algorithm 1) for the graph convolution layer is defined as $H^{(1)} = (\hat{A}H^{(0)}\theta^{(0)})$. $H^{(1)}$ is matrix of activations of first layer, $H^{(0)} = X$, $\theta^{(0)}$ is a trainable weight matrix of layer 0 and is a non-linear activation function.
3. Feed the convoluted representations of the vertices $\hat{A}H^{(0)}$ are fed to a standard fully connected layer.
4. Apply the sigmoid function row-wise on the fully connected last layer in the GCN.
5. Compute the cross entropy loss on known node labels.
6. Back-propagate the loss and update the weight matrix $\theta^{(0)}$.

Convoluted representation of node is produced by propagation rule by aggregating the labeled and unlabeled neighbors of the node. During training, back-propagation of the supervised binary cross-entropy loss leads to update the weights $\theta^{(0)}$ shared across all nodes. This loss depends on the latent feature representations of labeled nodes, which in turn depends on labeled and unlabeled nodes. Thus the learning becomes semi-supervised.

Algorithm 1: Propagation rule for the graph convolution layer. Adapted from [57]

In : Graph $(,)$; input features $\{x_v, \forall v \in\}$; depth K; weight matrices $\{W^k, \forall k \in [1, K]\}$; non-linearity σ; differentiable aggregator functions $\{\text{AGGREGATE}_k, \forall k \in [1, K]\}$; neighborhood function $\mathcal{N} : v \rightarrow 2$

Out Vector embedding z_v for all $v \in$

:

$h_v^0 \leftarrow x_v, \forall v \in ;$

for $k = 1...K$ **do**

  for $v \in$ **do**

    $h^k_{\mathcal{N}(v)} \leftarrow \text{AGGREGATE}_k(\{h_u^{k-1}, \forall u \in \mathcal{N}(v)\});$

    $h_v^k \leftarrow \sigma\left(W^k \cdot \text{COMBINE}(h_v^{k-1}, h^k_{\mathcal{N}(v)})\right)$

  end

  $h_v^k \leftarrow \text{NORMALIZE}(h_v^k), \forall v \in$

end

$z_v \leftarrow h_v^K, \forall v \in$

4.2 Relation of proposed model with laplacian smoothing

The proposed method performs operations that are equivalent to a laplacian smoothing operation. Consider a curve with points $p_i, p_{i-1}, p_{i+1}..$ as shown in Figure 6 where laplacian smoothing is applied to bring each point closer to weighted average of its neighbors using Eq. 1 and 2.

$$p_i \leftarrow p_i + \frac{1}{2}L(p_i) \tag{1}$$

$$L(p_i) = \frac{p_{i+1} + P_{i-1}}{2} = p_i \tag{2}$$

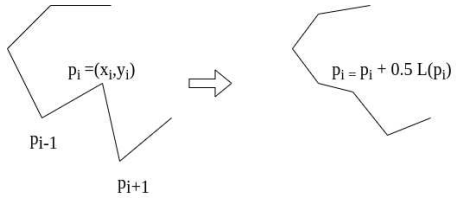

Fig. 6: Example of laplacian smoothing where a curve is shown before and after the smoothing operation is applied [58]

For a graph $G = (V, E)$ with adjacency matrix A such that $a_{ij} = 1, |i - j| = 1$ and degree matrix $D = diag(d_1, d_2, .., d_n)$, the smoothing operation can be re-written as,

$$P \leftarrow (I - \frac{1}{2} L_{rw}) P \tag{3}$$

where,

$$P = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_n & y_n \end{bmatrix} \quad L_{\text{rw}} = \begin{bmatrix} 1 & -1 & & & \\ -\frac{1}{2} & 1 & -\frac{1}{2} & & \\ & \ddots & \ddots & \ddots & \\ & & -\frac{1}{2} & 1 & -\frac{1}{2} \\ & & & -1 & 1 \end{bmatrix} \tag{4}$$

The matrix $L = D - A$ is the normalized graph Laplacian and has two versions $L_{sym} = D^{-1/2} L D^{-1/2}$ and $L_{rw} = D^{-1} L$. The laplacian smoothing operation is $P \leftarrow (I - \gamma L_{rw}) P$ and $0 \leq \gamma \leq 1$ controls strength of smoothing. Setting $\gamma = 0$, laplacian smoothing becomes equivalent to non-linear mapping function (identity function) being learned by GCN. The Laplacian smoothing computes the local average of each vertex as its new representation. After the smoothing over node neighborhoods, nodes that share neighbors tend to have similar feature representations or labels.

4.3 Complexity analysis

For an input graph with N nodes and feature vectors of length C, the adjacency matrix A will be in R^{N*N}, and the input feature matrix V^a will be of size R^{N*C}. For a given graph convolution layer, to obtain an output $V^{out} = R^{N*F}$, the time complexity is $O(N^2 CF)$.

5 Experimental study

Experiments to evaluate the efficacy of proposed model with others are elaborated (see Section 5.1) along with performance metrics (see Section 5.2). The results of the comparative study are discussed in Section 5.3.

5.1 Description of experiment

Proposed model is evaluated on twelve datasets of the Bitcoin Transaction graph (one for each year from 2009-2020). The graph datasets were split in 8:1:1 for training, test, and development set. Hyper-parameters of the models are listed, and scripts, datasets, and notes for reproducibility are available at Github repo [5]. Model-IIIA and IIIB were implemented using the StellarGraph Python library, and Models-I, II were implemented using the XGBOOST Python library.

List of model hyper-parameters:

- Model-I:
  1. layer-1 size=16
  2. layer-2 size=16
  3. layer-1 activation=relu
  4. layer-2 activation=relu
  5. dropout=0.5
  6. optimizer=adam
  7. loss=categorical cross-entropy
  8. epochs=200
  9. early stopping patience=50
  10. restore-best-weights
- Model-II:
  1. learning_rate = 0.0001
  2. n_estimators = 100
  3. max_depth = 1
  4. colsample_bylevel = 1
  5. colsample_bytree = 1
  6. subsample = 1
- Model-IIIA and IIIB:
  1. learning_rate = 0.0001
  2. n_estimators = 100
  3. max_depth = 1
  4. colsample_bylevel = 1
  5. colsample_bytree = 1
  6. subsample = 1

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 $(\hat{y}_i, y_i)$, Accuracy (A) is defined by Eq. 11.

$$Accuracy(A) = \frac{t_p + t_n}{t_p + t_n + f_p + f_n} \tag{5}$$

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

Loss (see Eq. 12) was calculated over m observations in the dataset as the average of loss for each observation i given the prediction $\hat{y}^{(i)}$ and actual label $y^{(i)}$.

$$Loss = \frac{1}{m}\sum_{i=1}^{m} L(\hat{y}^{(i)}, y^{(i)}) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}log(\hat{y}^{(i)}) - (1-y^{(i)})log(1-\hat{y}^{(i)})] \quad (6)$$

5.3 Experimental results and discussion

Accuracy of the Models-I, II, IIIA, and IIIB on the classification task is given in Table 11. Accuracy on the datasets 2012-2020 is below 50% as the classification targets in the dataset increase.

Table 11: Accuracy in Bitcoin transaction dataset

	Model-I			**Model-II**			**Model-IIIA**			**Model-IIIB**		
	Train	**Test**	**Dev**	**Train**	**Test**	**Dev**	**Train**	**Test**	**Dev**	**Train**	**Test**	**Dev**
2009	-	-	-	-	-	-	-	-	-	-	-	-
2010	0.6647	0.6684	0.6532	0.9033	0.8859	0.90	0.889	0.891	0.8898	0.92	0.91	0.9216
2011	0.8365	0.8342	0.8306	0.8678	0.8660	0.8677	0.8575	0.8541	0.8575	0.8678	0.8544	0.8677
2012	0.4894	0.5059	0.5076	0.5566	0.5565	0.5564	0.5596	0.5591	0.5595	0.5596	0.5445	0.5595
2013	0.3491	0.353	0.3483	0.3732	0.3752	0.3761	0.4247	0.4193	0.4198	0.3997	0.3898	0.3918
2014	0.0936	0.0667	0.0670	0.3259	0.3251	0.3251	0.3962	0.3957	0.3251	0.3962	0.3957	0.3961
2015	0.2066	0.205	0.2052	0.2857	0.2873	0.2852	0.3439	0.3411	0.2852	0.3651	0.3673	0.3650
2016	0.2118	0.2719	0.2712	0.3236	0.3256	0.3227	0.3517	0.3525	0.3516	0.3676	0.3684	0.3674
2017	0.2355	0.2377	0.2358	0.2617	0.2618	0.2617	0.3451	0.3450	0.3448	0.3484	0.3482	0.3481
2018	0.1738	0.1667	0.1656	0.2269	0.2280	0.2268	0.2269	0.2280	0.2268	0.2269	0.2280	0.2268
2019	0.1263	0.2002	0.1106	0.3331	0.3319	0.3339	0.3661	0.3651	0.3682	0.3737	0.3739	0.3767
2020	0.5163	0.5167	0.5154	0.5166	0.5149	0.5165	0.5166	0.5149	0.5165	0.5166	0.5149	0.5165

Performance of Models-IIIA and IIIB is 20-25% higher than Models-I and II. Hence, it is concluded that the performance of the classification task is improved by using GCN-based features and transaction features. Misclassification loss of the proposed Models-I, II, IIIA, and IIIB on the classification task is given in Table 12. As hyper-parameter tuning for the models was not performed, loss on the development set is not given in Table 12. The loss of Models-II, IIIA, and IIIB is lower than Model-I on all datasets except 2010. Hence, the accuracy was higher.

Table 12: Loss in Bitcoin transaction dataset

	Model-I		**Model-II**		**Model-IIIA**		**Model-IIIB**	
	Train	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**
2009	-	-	-	-	-	-	-	-
2010	7.9393e-08	0.0000	0.3	0.31	0.25	0.24	0.22	0.22
2011	2.725	2.67	0.49	0.5	0.32	0.34	0.32	0.33
2012	8.2191	7.9637	1.13	1.14	0.84	0.8	0.73	0.76
2013	10.4036	10.4281	1.64	1.65	1.34	1.34	1.23	1.24
2014	14.5889	15.0427	2.01	2.01	1.4	1.41	1.41	1.41
2015	12.71	12.81	1.95	1.96	1.43	1.43	1.37	1.37
2016	12.67	11.73	1.87	1.9	1.24	1.22	1.03	1.06
2017	12.1663	12.2868	1.96	2.01	1.23	1.27	1.09	1.13
2018	13.0660	13.4304	1.98	1.98	1.52	1.53	1.4	1.41
2019	13.9138	12.89	1.99	1.99	1.46	1.46	1.34	1.34
2020	7.7786	7.7894	1.3752	1.38	1.39	1.39	1.36	1.37

The utilization of RAM of the proposed Models-I, II, IIIA, and IIIB on the classification task is given in Table 13. The requirement of RAM is below 10GB, making it suitable for deployment in a desktop or mobile device.

Table 13: Utilization of RAM by Models

	Model-I	**Model-II**	**Model-IIIA**	**Model-IIIB**
2009	-	-	-	-
2010	3.1	0.8	0.9	0.9
2011	3.1	0.8	0.9	0.9
2012	3.5	0.99	1.02	1.04
2013	3.7	1.7	1.7	2.01
2014	4.6	1.2	1.96	2.62
2015	5.6	3.02	3.43	9.75
2016	10.4	4.2	8.3	8.2
2017	6.9	5.36	5.38	7.2
2018	5	3.29	3.35	3.66
2019	4.1	1.47	2.7	3.27
2020	3.1	1.69	2.07	2.15

To improve performance on datasets 2012-2020, the classification task was modified from multi-class to binary (legal or illegal) by merging legal and illegal labels to a single class. Illegal labels merged were dark market, blacklist, gambling, criminals, mixer, Ponzi, ransom, sextort, laundering, scams, and bomb. Similarly, legal labels merged together were pools, exchange, trading, paymentgateway, wallets, p2plender, faucet, donations, p2pmarkets, unclassified, videosharing, miner, bond, explorer, cybersec, affiliatemarketing, and microworker. Table 14 gives accuracy of Models-I, II, IIIA and IIIB on the datasets.

Table 14: Accuracy in Bitcoin transaction dataset

	Model-I			**Model-II**			**Model-IIIA**			**Model-IIIB**		
	Train	**Test**	**Dev**	**Train**	**Test**	**Dev**	**Train**	**Test**	**Dev**	**Train**	**Test**	**Dev**
2012	0.8201	0.8185	0.8310	0.82	0.82	0.79	0.85	0.85	0.6	0.82	0.82	0.79
2013	0.7069	0.7095	0.6970	0.71	0.71	0.59	0.71	0.71	0.56	0.71	0.71	0.58
2014	0.7471	0.7509	0.7605	0.75	0.75	0.49	0.75	0.75	0.55	0.75	0.75	0.55
2015	0.7471	0.7509	0.7605	0.63	0.63	0.55	0.63	0.63	0.59	0.63	0.63	0.59
2016	0.7453	0.7485	0.7620	0.79	0.79	0.62	0.79	0.79	0.59	0.79	0.79	0.62
2017	0.6111	0.6234	0.6445	0.65	0.65	0.54	0.67	0.67	0.62	0.67	0.67	0.62
2018	0.6353	0.6366	0.6350	0.64	0.64	0.58	0.64	0.64	0.64	0.64	0.64	0.64
2019	0.5771	0.5792	0.5875	0.6	0.59	0.64	0.58	0.58	0.6	0.6	0.59	0.64
2020	0.3830	0.3843	0.3960	0.62	0.62	0.52	0.62	0.62	0.51	0.62	0.62	0.52

Models-IIIA and IIIB use additional features such as GCN based features and transaction features to improve accuracy on the dataset by 20-30% compared to Models-I and II. Overall, the accuracy of binary classification has improved compared to multi-class classification by 45-55%. Misclassification loss of the proposed Models-I, II, IIIA, and IIIB on the classification task is given in Table 15.

Table 15: Loss in Bitcoin transaction dataset

	Model-I			**Model-II**			**Model-IIIA**			**Model-IIIB**		
	Train	**Test**	**Dev**	**Train**	**Test**	**Dev**	**Train**	**Test**	**Dev**	**Train**	**Test**	**Dev**
2012	9.7758e-08	9.5563e-08	9.9063e-08	0.69	0.69	0.68	0.68	0.69	0.69	0.82	0.82	0.79
2013	8.4269e-08	8.4580e-08	8.3089e-08	0.69	0.69	0.68	0.69	0.69	0.68	0.69	0.69	0.68
2014	8.9061e-08	8.9510e-08	9.0659e-08	0.69	0.69	0.68	0.69	0.69	0.68	0.69	0.69	0.68
2015	8.9061e-08	8.9510e-08	9.0659e-08	0.69	0.69	0.68	0.69	0.69	0.68	0.69	0.69	0.68
2016	8.8847e-08	8.9234e-08	9.0837e-08	0.69	0.69	0.68	0.69	0.69	0.68	0.69	0.69	0.68
2017	6.2395	6.0705	5.730	0.69	0.69	0.68	0.69	0.69	0.68	0.69	0.69	0.68
2018	7.5734e-08	7.5893e-08	7.5698e-08	0.69	0.69	0.68	0.69	0.69	0.68	0.69	0.69	0.68
2019	6.8031	6.7825	6.6487	0.69	0.69	0.68	0.69	0.69	0.68	0.69	0.69	0.68
2020	4.5651e-08	4.5810e-08	4.7207e-08	0.69	0.69	0.68	0.69	0.69	0.68	0.69	0.69	0.68

With a lower loss in binary classification compared to multi-class classification, the accuracy of the task improves. The utilization of RAM of the proposed Models-I, II, IIIA, and IIIB on the classification task is given in Table 16. Lower RAM is consumed for binary classification compared to multi-class classification.

Table 16: Utilization of RAM by Models

	Model-I	**Model-II**	**Model-IIIA**	**Model-IIIB**
2012	1.2	1.13	1.18	1.26
2013	1.6	1.33	1.68	1.68
2014	4.6	1.43	1.57	1.8
2015	5.9	3.3	3.2	4
2016	9.7	8.4	8.5	8.4
2017	12.1	9.1	9.3	9.7
2018	12.7	1.4	1.62	1.97
2019	13.8	2.9	3.2	3.5
2020	15.5	1.34	1.46	1.89

Based on the performance of the two models viz. multi-class and binary classification as per the requirement, an appropriate model can be selected. For fast and accurate decisions, binary classification is appropriate, and for pinpointing category of bitcoin users, a multi-class classifier is suitable.

5.4 Summary of Results

- Performance of Models-IIIA and IIIB is 20-25% improved than Models-I and II by using GCN based features along with transaction features.
- Utilization of RAM of the proposed Models-I, II, IIIA, and IIIB is below 10GB making them suitable for deployment in a desktop or mobile device
- accuracy of binary classification has improved compared to multi-class classification by 45-55%
- Lower RAM is consumed for binary classification compared to multi-class classification by 55-75%.

6 Conclusion and Future works

Bitcoin forensic tools rely on artificial intelligence for tracking illegal and legal transactions on Blockchain. To resolve the challenge due to high volume of transactions, the current paper proposes deep learning based graph neural network model using spectral graph convolutions and transaction features for identifying illegal transactions and labeling the transactions by their originator type. The model leverage the transaction graph of Bitcoin and features of transactions. It was observed through experiments that supervised learning is challenging due to the diverse types of entities present on the Bitcoin network. The classifier faced difficulty in identifying discriminative features from the data. The classification task was divided into identifying whether transactions were legal or illegal (binary classification) and classifying transactions to one of the twenty-eight types of users (multi-class classification).

On the multi-class classification task, the proposed model obtained accuracy higher than existing models as given: Model-I (12-83%), Model-II (22-90%), Model-IIIA (22-89%), and Model-IIIB (22-92%). On the binary classification task, the proposed model obtained accuracy as given: Model-I (60-82%), Model-II (60-85%), Model-IIIA (60-85%), and Model-IIIB (59-82%). From the performance of the models, it was observed that Model-I and II had lower performance, which was improved by proposed model. Concluding that models' performance was improved by using GCN based features along with transaction features. The performance improvement also proves that vanilla models i.e., models using only GCN or transaction features, will have comparatively lower performance than models that use GCN based features along with transaction features. Based on the performance of the two models viz. multi-class and binary classification as per the requirement, an appropriate model can be selected. For fast and accurate decisions, binary classification is appropriate, and for pinpointing the category of bitcoin users, the multi-class classifier is suitable.

Feature engineering additional features from Blockchain to improve accuracy is proposed in future works. To encourage further exploration in bitcoin illegal transaction detection, the datasets and scripts are openly accessible on the Github repository.

Compliance with Ethical Standards

The authors declare that they have complied with ethical standards of the journal during their research.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have influenced the work reported in this paper.

References

1. Farida Sabry, Wadha Labda, Aiman Erbad, Husam Al Jawaheri, and Qutaibah Malluhi. Anonymity and privacy in bitcoin escrow trades. In *Proceedings of the 18th ACM Workshop on Privacy in the Electronic Society*, pages 211–220, 2019.
2. Mauro Conti, E Sandeep Kumar, Chhagan Lal, and Sushmita Ruj. A survey on security and privacy issues of bitcoin. *IEEE Communications Surveys & Tutorials*, 20(4):3416–3452, 2018.
3. John Bohannon. The bitcoin busts, 2016.
4. Angela SM Irwin and Adam B Turner. Illicit bitcoin transactions: challenges in getting to the who, what, when and where. *Journal of money laundering control*, 2018.
5. Adam Turner and Angela Samantha Maitland Irwin. Bitcoin transactions: a digital discovery of illicit activity on the blockchain. *Journal of Financial Crime*, 2018.
6. Mohamed Rahouti, Kaiqi Xiong, and Nasir Ghani. Bitcoin concepts, threats, and machine-learning security solutions. *IEEE Access*, 6:67189–67205, 2018.
7. Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. Technical report, Manubot, 2019.
8. Sehyun Park, Seongwon Im, Youhwan Seol, and Jeongyeup Paek. Nodes in the bitcoin network: comparative measurement study and survey. *IEEE Access*, 7:57009–57022, 2019.
9. Joseph Bonneau, Arvind Narayanan, Andrew Miller, Jeremy Clark, Joshua A Kroll, and Edward W Felten. Mixcoin: Anonymity for bitcoin with accountable mixes. In *International Conference on Financial Cryptography and Data Security*, pages 486–504. Springer, 2014.
10. Yining Hu, Suranga Seneviratne, Kanchana Thilakarathna, Kensuke Fukuda, and Aruna Seneviratne. Characterizing and detecting money laundering activities on the bitcoin network. *arXiv preprint arXiv:1912.12060*, 2019.

11. Rainer Böhme, Nicolas Christin, Benjamin Edelman, and Tyler Moore. Bitcoin: Economics, technology, and governance. *Journal of economic Perspectives*, 29(2):213–38, 2015.
12. Víctor Gabriel Reyes-Macedo, Moisés Salinas-Rosales, and Gina Gallegos Garcia. A method for blockchain transactions analysis. *IEEE Latin America Transactions*, 17(07):1080–1087, 2019.
13. Masarah Paquet-Clouston, Matteo Romiti, Bernhard Haslhofer, and Thomas Charvat. Spams meet cryptocurrencies: Sextortion in the bitcoin ecosystem. In *Proceedings of the 1st ACM Conference on Advances in Financial Technologies*, pages 76–88, 2019.
14. Kentaroh Toyoda, P Takis Mathiopoulos, and Tomoaki Ohtsuki. A novel methodology for hyip operators' bitcoin addresses identification. *IEEE Access*, 7:74835–74848, 2019.
15. Chaehyeon Lee, Sajan Maharjan, Kyungchan Ko, and James Won-Ki Hong. Toward detecting illegal transactions on bitcoin using machine-learning methods. In Zibin Zheng, Hong-Ning Dai, Mingdong Tang, and Xiangping Chen, editors, *Blockchain and Trustworthy Systems*, pages 520–533, Singapore, 2020. Springer Singapore.
16. Sean Foley, Jonathan R Karlsen, and Tālis J Putniņš. Sex, drugs, and bitcoin: How much illegal activity is financed through cryptocurrencies? *The Review of Financial Studies*, 32(5):1798–1853, 2019.
17. Tengyu Liu, Jingguo Ge, Yulei Wu, Bowei Dai, Liangxiong Li, Zhongjiang Yao, Jifei Wen, and Hongbin Shi. A new bitcoin address association method using a two-level learner model. In Sheng Wen, Albert Zomaya, and Laurence T. Yang, editors, *Algorithms and Architectures for Parallel Processing*, pages 349–364, Cham, 2020. Springer International Publishing.
18. Francesco Zola, Maria Eguimendia, Jan Lukas Bruse, and Raul Orduna Urrutia. Cascading machine learning to attack bitcoin anonymity. In *2019 IEEE International Conference on Blockchain (Blockchain)*, pages 10–17. IEEE, 2019.
19. Wei Shao, Hang Li, Mengqi Chen, Chunfu Jia, Chunbo Liu, and Zhi Wang. Identifying bitcoin users using deep neural network. In Jaideep Vaidya and Jin Li, editors, *Algorithms and Architectures for Parallel Processing*, pages 178–192, Cham, 2018. Springer International Publishing.
20. Marc Jourdan, Sebastien Blandin, Laura Wynter, and Pralhad Deshpande. Characterizing entities in the bitcoin blockchain. In *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 55–62. IEEE, 2018.
21. Bruno Bogaz Zarpelão, Rodrigo Sanches Miani, and Muttukrishnan Rajarajan. Detection of bitcoin-based botnets using a one-class classifier. In Olivier Blazy and Chan Yeob Yeun, editors, *Information Security Theory and Practice*, pages 174–189, Cham, 2019. Springer International Publishing.
22. Yan Wu, Anthony Luo, and Dianxiang Xu. Identifying suspicious addresses in bitcoin thefts. *Digital Investigation*, 31:200895, 12 2019.
23. Y. Wu, F. Tao, L. Liu, J. Gu, J. Panneerselvam, R. Zhu, and M. N. Shahzad. A bitcoin transaction network analytic method for future blockchain forensic investigation. *IEEE Transactions on Network Science and Engineering*, pages 1–1, 2020.

24. Mark Weber, Giacomo Domeniconi, Jie Chen, Daniel Karl I Weidele, Claudio Bellei, Tom Robinson, and Charles E Leiserson. Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics. *arXiv preprint arXiv:1908.02591*, 2019.
25. Hao Hua Sun Yin, Klaus Langenheldt, Mikkel Harlev, Raghava Rao Mukkamala, and Ravi Vatrapu. Regulating cryptocurrencies: a supervised machine learning approach to de-anonymizing the bitcoin blockchain. *Journal of Management Information Systems*, 36(1):37–73, 2019.
26. Lingxiao Yang, Xuewen Dong, Siyu Xing, Jiawei Zheng, Xinyu Gu, and Xiongfei Song. An abnormal transaction detection mechanim on bitcoin. In *2019 International Conference on Networking and Network Applications (NaNA)*, pages 452–457. IEEE, 2019.
27. Zhen Zhang, Tianyi Zhou, and Zhitong Xie. Bitscope: Scaling bitcoin address de-anonymization using multi-resolution clustering.
28. S. Phetsouvanh, F. Oggier, and A. Datta. Egret: Extortion graph exploration techniques in the bitcoin network. In *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 244–251, 2018.
29. M. Bartoletti, B. Pes, and S. Serusi. Data mining for detecting bitcoin ponzi schemes. In *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*, pages 75–84, 2018.
30. Kota Kanemura, Kentaroh Toyoda, and Tomoaki Ohtsuki. Identification of darknet markets' bitcoin addresses by voting per-address classification results. In *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 154–158. IEEE, 2019.
31. Cuneyt Gurcan Akcora, Yitao Li, Yulia R. Gel, and Murat Kantarcioglu. Bitcoinheist: Topological data analysis for ransomware detection on the bitcoin blockchain, 2019.
32. Rebecca S. Portnoff, Danny Yuxing Huang, Periwinkle Doerfler, Sadia Afroz, and Damon McCoy. Backpage and bitcoin: Uncovering human traffickers. In *KDD '17*, 2017.
33. Patrick Monamo, Vukosi Marivate, and Bheki Twala. Unsupervised learning for robust bitcoin fraud detection. In *2016 Information Security for South Africa (ISSA)*, pages 129–134. IEEE, 2016.
34. Patrick M Monamo, Vukosi Marivate, and Bhesipho Twala. A multifaceted approach to bitcoin fraud detection: Global and local outliers. In *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 188–194. IEEE, 2016.
35. Haohua Sun Yin and Ravi Vatrapu. 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)*, pages 3690–3699. IEEE, 2017.
36. Mikkel Alexander Harlev, Haohua Sun Yin, Klaus Christian Langenheldt, Raghava Mukkamala, and Ravi 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.

37. L. Nan and D. Tao. Bitcoin mixing detection using deep autoencoder. In *2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*, pages 280–287, 2018.
38. Jiaqi Liang, Linjing Li, Shu Luan, Lu Gan, and Daniel Zeng. Bitcoin exchange addresses identification and its application in online drug trading regulation. 2019.
39. Thai Pham and Steven Lee. Anomaly detection in bitcoin network using unsupervised learning methods. *arXiv preprint arXiv:1611.03941*, 2016.
40. Andreas Bogner. 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*, pages 5–8, 2017.
41. Fabio Aiolli, Mauro Conti, Ankit Gangwal, and Mirko Polato. Mind your wallet's privacy: Identifying bitcoin wallet apps and user's actions through network traffic analysis. 04 2019.
42. Jordi Zayuelas Muñoz. Detection of bitcoin miners from network measurements. B.S. thesis, Universitat Politècnica de Catalunya, 2019.
43. Jordi Herrera-Joancomartí. Research and challenges on bitcoin anonymity. In *Data Privacy Management, Autonomous Spontaneous Security, and Security Assurance*, pages 3–16. Springer, 2014.
44. Anil Gaihre, Yan Luo, and Hang Liu. Do bitcoin users really care about anonymity? an analysis of the bitcoin transaction graph. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 1198–1207. IEEE, 2018.
45. Andrea Pinna, Roberto Tonelli, Matteo Orrú, and Michele Marchesi. A petri nets model for blockchain analysis. *The Computer Journal*, 61(9):1374–1388, 2018.
46. Alex Greaves and Benjamin Au. Using the bitcoin transaction graph to predict the price of bitcoin. *No Data*, 2015.
47. Stefano Bistarelli, Ivan Mercanti, and Francesco Santini. A suite of tools for the forensic analysis of bitcoin transactions: Preliminary report. In *European Conference on Parallel Processing*, pages 329–341. Springer, 2018.
48. Dmitry Ermilov, Maxim Panov, and Yury Yanovich. Automatic bitcoin address clustering. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 461–466. IEEE, 2017.
49. Marie Vasek and Tyler Moore. There's no free lunch, even using bitcoin: Tracking the popularity and profits of virtual currency scams. In Rainer Böhme and Tatsuaki Okamoto, editors, *Financial Cryptography and Data Security*, pages 44–61, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.
50. Damiano Di Francesco Maesa, Andrea Marino, and Laura Ricci. The bow tie structure of the bitcoin users graph. *Applied Network Science*, 4(1):56, 2019.
51. Damiano Di Francesco Maesa, Andrea Marino, and Laura Ricci. The graph structure of bitcoin. In *International Conference on Complex Networks and their Applications*, pages 547–558. Springer, 2018.
52. Damiano Di Francesco Maesa, Andrea Marino, and Laura Ricci. Data-driven analysis of bitcoin properties: exploiting the users graph. *International Journal of Data Science and Analytics*, 6(1):63–80, 2018.

53. Damiano Di Francesco Maesa, Andrea Marino, and Laura Ricci. Uncovering the bitcoin blockchain: an analysis of the full users graph. In *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 537–546. IEEE, 2016.
54. Israa Alqassem, Iyad Rahwan, and Davor Svetinovic. The anti-social system properties: Bitcoin network data analysis. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2018.
55. A Janda. Walletexplorer. com: Smart bicoin block explorer, 2016.
56. Emmanuelle Anceaume, Thibaut Lajoie-Mazenc, Romaric Ludinard, and Bruno Sericola. Safety analysis of bitcoin improvement proposals. In *2016 IEEE 15th International Symposium on Network Computing and Applications (NCA)*, pages 318–325. IEEE, 2016.
57. William L Hamilton, Rex Ying, and Jure Leskovec. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584*, 2017.
58. Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.