Mixers Detection in bitcoin network: a step towards detecting money laundering in crypto-currencies

M. Mazhar Rathore*, Sushil Chaurasia, Dhirendra Shukla*
Dr. J. Herbert Smith Centre
University of New Brunswick
Fredericton, New Brunswick, Canada
{rathore.mazhar, sushil.chaurasia, dshukla}@unb.ca

2022 IEEE International Conference on Big Data (Big Data) | 978-1-6654-8045-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/BigData55660.2022.10020982

***Abstract*—Anonymity is one of major factors that is causing the rise of bitcoin crypto-currency. There are several attacks (positive or negative) to de-anonymize the bitcoin addresses, in order to link the bitcoin entity to a physical entity or person. Bitcoin mixing service (called mixer) is one of the approaches to keep the user's crpto-anonymity in the transparent ledger of bitcoin network. Mixers breaks the link between the sender and the receiver by mixing up coins received from multiple sources, while creating a mess to make it impossible to identify the actual sender of bitcoins. On the other hand, mixing services are being vastly exploited by criminals for laundering the illegal money, taken from frauds, ransom, scams, or other illegal activities. Detecting mixing services or mixer's involvement in a transaction can help in discovering money laundering activities in the bitcoin blockchain. Existing mixer's detection approaches either have a low accuracy-rate due to the changing nature of the mixing process or they are not efficient enough to be implemented in a real-time environment. In this paper, we developed a highly accurate decision-tree based model using C4.5 machine learning approach to identify addresses providing mixing services. To make this detection process efficient and be able to work in a real environment, we reduced overall feature-set to only eight features, minimizing overall computation time. Further, we shrink the decision-tree using reduced error-pruning to make the detection process faster. With the short decision-tree-size of 55 nodes, we achieved the accuracy of more than 97%, which is quite higher.**

***Index Terms*—money laundering, bitcoin analytics, mixers detection, bitcoin address classification.**

I. Introduction

The high level of anonymity that bitcoin offers to its users is one of its distinguishing features [1]. Since Nakamoto's initial proposal of bitcoin in 2008, both researchers and investors have given it considerable attention to make it as the first widely used open source digital currency with no central body in charge of regulating or managing its supply [2], [3]. Because to its anonymity, lack of a centralised authority, and high potential for profit, people use the bitcoin for a variety of activities, including payment, commerce, and investment. On the other side, due to anonymity feature in bitcoin, variety of crimes occur, including hired killings, sponsorship of terrorism, trafficking of drugs, computer hacking, weapons, forgeries, illegal gambling, ponzi schemes, money laundering, illegal mining, the propagation of blatant theft, and ransomware [4]–[6].

The current work is supported by Atlantic Innovation Fund and MITACS (IT24468).

Efforts have been made to de-anonymize bitcoin addresses and link a bitcoin entity to a real-world entity. Although, most of these efforts are for positive reasons, to cater above-mentioned illegal activities, but they are considered as attacks on bitcoin anonymity feature. Therefore, contributors to the bitcoin network are establishing services, like mixers or tumblers, to counter these attacks. Mixer is considered as one of the most deceptive techniques for hiding the sender and recipient of the coins in cryptocurrency [7]. The goal of mixing services is to restore network anonymity. By leveraging built-in features of both blockchain and bitcoin, bitcoin mixing services give their consumers more anonymity. The primary objective of bitcoin mixing services is to separate bitcoins from their frequently used source. It is almost hard to track back mixed bitcoin to their contaminated source when a bitcoin mixer is in use. In the blockchain, the consumer may verify the legitimacy of the bitcoins they have received (using some servers such as blockchain.com). If the mixing of bitcoins is done properly, there is no connection (or "zero percent taint") between the bitcoins that were deposited and those that were received.

A thorough study on various common mixing services from various perspectives was conducted by Pakki et al. [7]. Although this study does not take into account the functional behaviour of mixers, it is nevertheless useful for understanding how mixing services operate. By the mixing service provider, a freshly created bitcoin address is typically given to consumers to deposit bitcoins. Taking bitcoins from several customers, the mixer distributes bitcoins from its reserve to customers given recipients. Mixer charges the fee for this service. The payouts are spaced out over time and some randomization is added to the split of money and/or the mixing fee to increase anonymity. Some bitcoin mixing providers provide a service to repeat users to prevent accidentally paying out to the same user any (already) deposited contaminated bitcoins in their reserve during a later usage of the mixing service. A returning customer number is given to the consumer following each "mix". When using the mixing service again, this number may be shown. Once the mixer is aware of which bitcoins in the reserve were previously deposited, it will refuse to provide the customer those particular bitcoins.

On one end, mixers provide security against the de-anonymization attack in the bitcoin. On the other end, mixing

services have considerably raised financial crimes, particularly money laundering. The link between the donor and the recipient of money is distorted when several sources of crypto-currency are combined with dirty money, making it difficult and obscure to track down filthy money [8], [9]. By mixing these earnings with other sources of money and interfering with the monitoring of these incomes, such services are frequently used to get rid of the trail of income from ransomware, thefts, sales of guns and narcotics, and other illicit activities.

Several studies have been done to identify mixing services (mixers or tumblers) in bitcoins. Some of them use the graph analysis [10]–[14], which is not a feasible way for real-time analysis of bitcoin addresses. AI and Machine learning is another powerful tool for classification problems [15], [16]. This is an another option for mixing service classification [17], [18], but existing machine learning-based approaches either have the problem of efficiency–as they use too many parameters for classification–or they have the low accuracy rate. Furthermore, with every passing day and advanced technology, mixing services are using new ways of hiding the sources, which are very challenging to identify.

To address the aforementioned challenges while detecting the mixers, in this paper, we designed an efficient machine learning-based model for mixing services detection. The model is based on C4.5 [19] decision tree leaning and utilizes only eight address-related features. In addition, as a prior study, we developed a bitcoin blockchain graph using Neo4j[1] and graphically analyse those transactions where mixers are involved. With graph analysis, we selected some statistical parameters related to transaction-patterns for machine learning modeling. Overall, our significant contributions are as follows:

- A bitcoin blockchain graph is developed in Neo4j to analysis the graphical patterns of mixers.
- Using the graph data analysis, we discovered several features that help in identifying transaction patterns.
- 36 bitcoin address-related features are selected based on thorough analysis of mixing services, and then these features are extracted and computed using an external API *blockchain.com*.
- Next, the feature set is reduced to eight most-important features using *information gain*, reducing the computation time and achieving higher accuracy.
- Finally, a decision-tree based model is developed using C4.5 machine learning algorithm for mixers detection.
- The designed model is thoroughly evaluated with different perspective for correctness and efficiency. We achieved 97% accuracy with the shortest possible tree of 55 nodes and a feature set of size 8 for each address-sample.

The remainder of this paper is structured as follows. Literature review is investigated in Section II. The preliminaries concepts, including the bitcoin blockchain overview, *information gain*, C4.5 learning algorithm, are discuss in Section III. The discussion on the overall machine learning process, features extraction and selection, and the designed model is presented in Section IV. The evaluation results are illustrated in Section V. Last section concludes our work.

[1]https://neo4j.com/

II. Related work

Recent comprehensive research on various common mixing services from various perspectives was conducted by Pakki et al. [7]. Although this study does not take into account the functional behaviour of mixers, it is nevertheless useful for understanding how mixing services operate. We know that detecting bitcoin mixers is a major factor to consider during bitcoin laundering analysis [7], [20]. There are several publications that focus on bitcoin transaction and address analysis, aimed at frauds detection, address deanonymization, mixers detection, and entity identification.

Finding fraudulent addresses is essential to protect financial systems. For locating malicious addresses in the bitcoin network, studies can be categorised into two groups. Finding unusual transactions and users comes first. The subsequent priority is to concentrate on certain illegal websites, such as scams, darknet markets, ransomware, and hacks. Primary social network strategies are put out by [1] for the purpose of discovering potentially anomalous transactions and individuals in the bitcoin transaction network. To find irregularities in the bitcoin transaction network, Monamo et al., [21] employed using unsupervised learning techniques. Lin et al., [18] suggested using a supervised classification model to identify addresses on the bitcoin network that are out of the ordinary behaviour.

In addition, Using the graph technology is another alternative to analyse the relationship between entities [22], [23]. So, the graph can be a straightforward way to investigate bitcoin transactions flow. For financial forensics, [10], [11] proposed graph convolutional network analysis.Also Ron et al., [11] investigated the characteristics of transaction graphs and grouped addresses that may be associated with the same entity. On the other hand, transaction graph analysis [12] can help to partially understand the corresponding input and output addresses that are confused by mixing services.

To connect addresses to entities, or collections of addresses owned by the same individuals or the same businesses, clustering techniques are offered by [2], [24]. The bitcoin network's graphs show the partial linkability between addresses and entities. Even while it is presently impossible to connect an arbitrary address to its owner in the actual world, the connected entities may be assessed using off-chain data like tags (mining pool, exchange wallet, etc.). Another method for analysing pattern in addition to graphs is motif. In directed hypergraphs, Ranshous et al., [13] introduces the notion of motifs, describing address exchange patterns. To identify bitcoin entity categories, then combines the graph-based feature motifs with address, entity, temporal, and centrality feature combinations [10], [14].

To identify mixing services, researchers used various pattern analysis techniques. The authors in [12] showed assaults against three various mixing services that were operational

in 2013. They first use taint analysis to identify the mixing techniques. They discovered that there were occasionally connections between sender and receiver even when the route between them had been cleaned in some instances. Due to the separation between the sender and the recipient, today's mixing services are far more clever and are thus impossible to identify via taint analysis. The authors of a white paper from the network security firm Novetta [25] specifically targeted and interacted with three mixing services from a structural standpoint. They came to the conclusion that the usage of mixers rendered taint analysis are useless. A second study by Chainalysis researchers Baltazar and Castro looked into three mixing services and attempted to identify certain trends [20] in those services. This study demonstrated how mixing services behave in a certain way, but they did not publicly disclose this behaviour. The method employed by [26] to identify mixing transactions, mixing addresses, sender addresses, and recipient addresses in the bitcoin network was statistical patterns analysis. They primarily used the functional features of three well-known mixing services to identify the transactions of those services. they sorted the addresses associated with those transactions into three categories: (1) sender addresses (also known as input transactions), (2) mixing service addresses (internal transactions), and (3) recipient addresses (or output transaction).

In addition to graph and statistical pattern analysis, transaction history can also serve an asset for address classification and mixing service detection using supervised machine learning. Based on supervised learning techniques, a set of attributes are presented in [17] to summarise the transaction history and identify addresses connected to HYIPs. These attributes have been expanded to recognise seven other categories of bitcoin-enabled services, including mixers [18]. Similarly, Yin and Vatrapu [27] uses supervised learning techniques on a database of tagged bitcoin addresses to categorise cyber-criminal groups. Using machine learning, a researchers' group [5] developed classifiers to identify bitcoin Ponzi scams. In [5], a sampling-based technique and a cost-sensitive strategy are taken into consideration simultaneously to address unbalanced data. By forecasting addresses that have not yet been detected, classifiers were trained using the synthetic minority over-sampling approach on unbalanced data to lessen bitcoin's anonymity [28], [29]. In contrast to earlier pattern-based approaches, Sun and Yang [30] claimed to have achieved a high recall technique when they suggested a method for mixture detection utilising long short-term memory (LSTM).

III. Preliminaries

A. Bitcoin blockchain

Bitcoin is the decentralised digital currency that can be transferred peer-to-peer without the involvement of any central party. Al-Farsi et al., [31] established a very easy and straight forward model to understand the blockchain process while considering bitcoin. Each user of the bitcoin have public and private keys-pair that is used to sign the transaction when sending bitcoins to any other user. The public key is also served as a address of a user to send and receive money. When a transaction is made by an address '$Addr_A$' to transfer bitcoins to another address '$Addr_B$', the transaction is singed by $Addr_A$. The transaction is broadcast over the peer-peer bitcoin network, where each miner node collects the transaction and add into its temporary block by verifying the signature on the transaction. Every miner is free to choose transactions to add into its temporary blocks, usually based on the offered mining fee by the sender. Each miner tries to mine its temporary block and if successfully mined, the miner announced it to the network with 'Proof of Work (PoW)'. All the nodes verify the mined PoW and add the block into the blockchain. all other nodes discard all those transactions from their temporary block which are already confirmed in the last mined block. Miner receives the mining fee against each of the added transaction from the sender and a mining reward from the network.

A single block have multiple transactions. The number of transactions in a block is purely the choice of miners. The more transactions in a block, it takes more time to mine the block but higher would be the reward. Each transaction has multiple inputs and outputs. An input in a transaction is the reference of a previous transaction in which the sender has received bitcoins from other address(s), and now in this transaction he wants to send those bitcoins to other address(s). The output in a transaction represents the receiver's address and the coin received by the receiving-address. A single transaction can have one or many senders and receivers. A bitcoin address may have multiple sending and receiving transactions. The transaction in which he sends the bitcoin (in this case, the address is in the inputs of the transaction), we called it input transaction, and the transaction where he got bitcoins, we called it output transaction of that address. An output transaction can only be used once as an input in any future transaction. If an output transaction is appeared in any of the future transactions as an input, it would be considered as spent transaction. For our study, we extracted basic parameters related to input and output transactions of each of the individual addresses and compute advanced parameters to train and build the model.

B. Information gain for parameter reduction

In decision trees, *information gain* (IG) is usually used to find out the best attribute to select as a node at the time of tree creation. In our case, we use IG to reduce the feature set. In the domain of machine learning classification, we can simply define the *information gain* of an attribute A is the amount of information we gain about a class-variable from observing attribute A. More formally, it can be defined as, IG of a training set T for an attribute A is the difference between a priori Shannon entropy $H(T)$ of the training set T and the conditional entropy $H(T|A)$.

For illustration, let T denotes a set of training samples. Each sample is in the form $(A, Y) = (A_1, A_2, A_3, ...A_k, Y)$, where A_i is the i^{th} attribute in the training set and Y is the label-variable. a_{ij} is the value of attribute A_i for sample j and y_j is

the corresponding label for sample j. The possible values of each attribute can be represented as $a_{ij} \in Domain(A_i)$ and the label as $y_j \in Domain(Y)$

The IG for an attribute A_i over the dataset T is defined in terms of Shannon entropy H as

$$IG(T, A_i) = H(T) - H(T|A_i) \quad (1)$$

where $H(T)$ is the Shannon entropy of dataset T with given labels y, and computed as

$$H(T) = - \sum_{v \in Domain(Y)} p_v \log p_v$$

p_v is the probability of a class-label v. $H(T|A_i)$ is the conditional entropy of training set with the given values of A_i and computed as

$$H(T|A_i) = \sum_{v \in Domain(A_i)} \frac{|S_{Ai}(v)|}{|T|} . H(S_{Ai}(v))$$

whereas, $S_{Ai}(v) = \{A \in T | A_i = v\}$, which means $S_{Ai}(v)$ is the set of all samples in T where $A_i = v$.

In this way, we compute the IG for each attribute in order to select the best attributes with highest IG.

C. C4.5 model and reduced-error pruning (REP)

C4.5 is a machine learning approach to generate a decision tree based on a given labeled training dataset. The algorithm is developed by Quinlan in 1993 [19]. It is the extension of his own ID3 algorithm [32]. With this approach the decision tree is build as follows. The algorithm start with three base cases, 1) if all the samples in the set belong to the same class, stop further extending that branch of tree and simple creates a leaf node to choose that class, 2) if no feature have any IG, creates a tree node higher up the tree using the expected value of the class, 3) if an instance of the previously unseen class comes, again creates a tree node higher up the tree using the expected value. Initially the base cases are checked and followed.Then, the normalised *information-gain* is computed for each of the attribute A (as discussed earlier in this section). Select the attribute A with the highest IG as a node. Do the split on A, and again find the best attribute using IG and add that attribute as a node. repetitively add nodes in this fashion until no split is possible. Sometime, the further split is possible but it reduces accuracy, in this case, we can also stop building tree.

Mostly, the constructed tree has the problem of overfitting. Pruning the tree after its creation may reduce the overfitting effect. In addition, it reduces the size of the tree to make the model more efficient. Reduced error pruning (REP) is considered simplest, widely used, and most effective way of tree pruning. With REP, we start with removing the leave nodes and replacing it with the most popular class, and by doing this, if the accuracy does not declined, we keep this change. We follow this approach for all the intermediate nodes as well. In case of removing an intermediate nodes, all the child nodes are also removed.

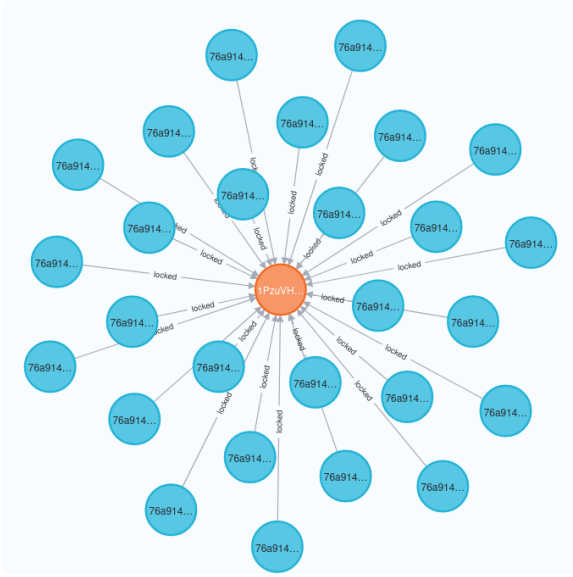

Fig. 1. Transactions of the Mixer "1PzuVH-grSH7rRJNttzgknuomMLohX54dCB".

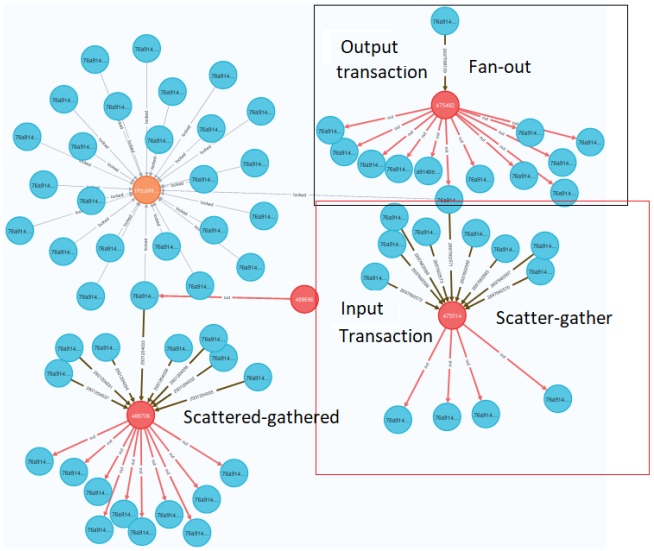

Fig. 2. Fan-out and scatter gathered transactions of the mixer "1PzuVH-grSH7rRJNttzgknuomMLohX54dCB"

IV. Proposed Model

A. Analysis

We performed thorough analysis on all transactions where a mixer is receiving or sending bitcoins. We build a bitcoin graph using Neo4j[2] and analysed patterns and flow of transactions. We found that most of the mixers transaction–where the mixer is sending or receiving bitcoins–have 'fan-in'/'fan-out' or 'scatter-gathered patterns'. In 'fan-in' pattern, the transaction has many senders and one receiver. Whereas, in the 'fan-out' pattern, there is one sender and many receivers in the transaction. In a 'scatter-gathered' (or 'gathered-scatter')

[2]neo4j.com

scenario, there is many to many relationship between number of inputs and outputs of the transaction. Analysis graphs on a mixer '$1PzuVHgrSH7rRJNttzgknuomMLohX54dCB$' are depicted in Fig. 1 and 2. Figure 1 shows all the bitcoins exchanges (represented by blue nodes) by the mixer (represented by the orange node). Each blue node stores the information of transaction(s) where the bitcoins are received and then sent to other addresses. A blue node can further be expanded to analyse the inputs and outputs of the transaction, as shown in Fig. 2.

The output transactions of a mixers are those transactions where bitcoins are received by the mixer (the mixer's address is included as one of the outputs of the transaction). Whereas, in an input transaction, the mixer sends the coins to other address(s) (the mixer's address is included as one of the inputs of the transaction). In Fig. 2, one of the mixer's exchange-node is extended to analyse the corresponding input transaction and output transaction of the mixer. The transactions are represented by red nodes, which have inputs and outputs. Inputs are represented by dark-black edges towards the node, and the outwards red-edges represent outputs of the transaction. The red node in the upper-right-block of the figure is the output transaction where the coins are received by the mixers. Whereas, the same received coins are then sent by the mixer in the next transaction, shown by red node in the down-right-block. Visualising these transactions from a mixer '$1PzuVHgrSH7rRJNttzgknuomMLohX54dCB$', the mixer mostly has 'fan-in', 'fan-out', and 'scatter gathered' types of transactions. You can see the output transaction in the upper-right part of the figure, the transaction is making a 'fan-out' pattern. Whereas in the lower part of the figure, the mixer–along with some other addresses–sends the coins to several other addresses in a 'scatter gathered' fashion. Due to such mixing behaviour, no one knows exactly which address is receiving the mixer's coins (in other words, hiding the source).

B. Parameter extraction and selection

At first, we extracted all the information for transactions and basic addresses-related parameters from *blockcypher.com*. We computed around 36 features from basic parameters for each of the addresses in the dataset. Computing 36 features and making decision based on them take a lot of time, which may not be feasible to do in a real-time environment, like bitcoin blockchain, where the blocks (with thousands of addresses) are being created after every few minutes. Therefore, we minimized the list of features and selected 10 best-features with highest *information gain*. Later, we reduced this number to 8 by choosing only one feature among strongly correlated features. The details of each of the selected 8 features with corresponding *information gain* are presented in Table I.

TABLE I
PARAMETERS SELECTED BY INFORMATION GAIN

Info. gain	Selected parameter	Details
.73	$freq\_per\_day$	Number-of-transactions/day
0.431	$mean\_spent\_input$	Average bitcoins-spent per spent transaction
0.404	$ratio\_AllPatterns$	Ratio of fan-in, fan-out, scatter gathered transactions to total transactions
0.39	$ratio\_InPattern$	Ratio of fan-in transactions to total transactions
0.346	$ratio\_received\_2\_0$	Ratio of no. of times the digit i in USD appeared in received transactions, where $i(10^3, 10^2, ..., 10^6)$, over no. of received transactions
0.303	$ratio\_spent$	Ratio of no. of spent transactions over total transactions
0.303	$ratio\_received$	Ratio of no. of received transactions over total transactions
0.293	$ratio\_spent\_2\_0$	ratio of no. of spent_2_0 over no. of spent transactions

C. Designed machine-learning model

To understand our approach of designing machine learning model for mixers-detection and then evaluating it, Fig. 3 presents the process flow. it is an ordinary machine learning approach where the labeled samples are given to the training function to build decision-tree (DT) using C4.5 learning algorithm. Once the DT is built, the unlabeled addresses and their corresponding features are passed to the DT-based model to split mixers and non-mixer addresses. The model's results are compared to the real address-labels to measure the correctness of the designed model.

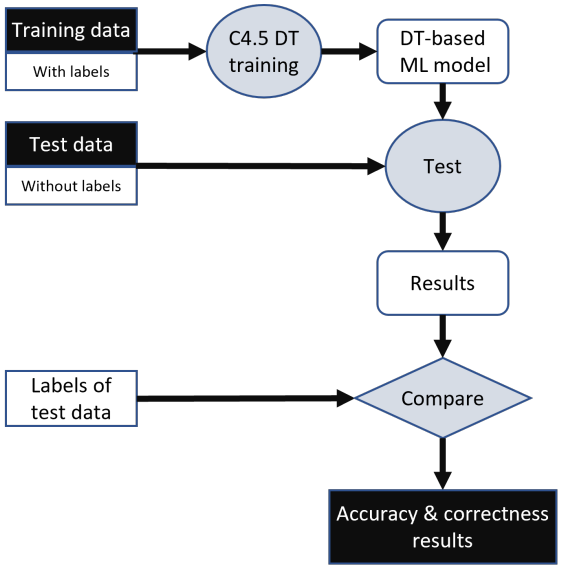

Fig. 3. The machine learning modeling approach

In a a practical bitcoin blockchain environment, our model can be deployed on a blockchain node to detect mixers in each of the latest mined-block. Figure 4 illustrates the deployment of our model in a practical-environment. At the time of new block addition in the bitcoin chain, our system extracts all the input and output addresses from each transaction and make a list of them. Against every address (that is never identified previously), all the features from Table I are computed. Next,

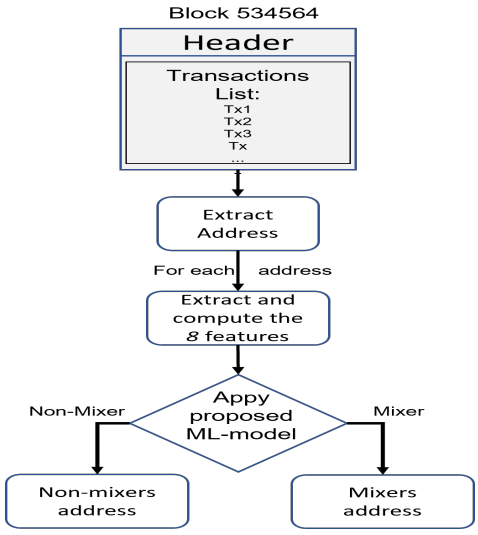

Fig. 4. The use of proposed model in real-environment

these features are given to our DT-based model, which filters the mixers addresses.

V. System Evaluation

A. Implementation environment and dataset

We evaluated our system in terms of accuracy and efficiency. For this purpose, we implemented the overall system on a Dell Precision 7920 machine with 'Ubuntu 20.04.5 LTS' operating system. The machine has 128GB RAM–although the memory usage is far less, around 1GB–and equipped with an Intel® Xeon(R) Gold 6230R CPU@2.10GHz x 52 processor. The machine has 52 processors but we used only one of them.

Regarding the dataset for training and testing, we leverage the mixers bitcoin addresses taken from [33], labeled by some clustering and heuristic techniques. The complete dataset has 26313 samples, including 3199 mixers and 23114 non-mixers addresses. To balance the dataset, we just used 6800 samples for training and testing (including 3199 mixers and 3601 non-mixer samples). Non-mixers addresses represent other services like exchange, faucet, HYIP, pool, market, and gambling. We used two versions of the dataset for evaluation; first one consist of 36 features corresponding to each address, whereas the second set is the reduced version, where only top 8 features are selected among 36 using information gain.

Furthermore, we used two approaches for our model evaluation. Initially, we built a full C4.5 tree on training data and test it for accuracy. Later, we reduced the full tree using reduced-error pruning (REP). We expect higher accuracy on full tree but lower efficiency, and higher efficiency on REP-tree but lower accuracy. However, our aim is to have very low accuracy loss due to pruning the tree. The sizes and the model building times for the full tree and REP-tree on both full set and the reduced set are presented in Table II. You can see a notable reduction in model-size due to pruning and data reduction.

TABLE II
DECISION-TREE DEPTH

Datasets	No. of leaves	Size of tree	Model-built time (sec)
Full dataset (without REP)	68	135	0.25
Full dataset (with REP)	42	83	0.22
Reduced dataset (without REP)	53	105	0.09
Reduced dataset(REP)	28	55	0.07

For address-related features-computation, we utilized block-cypher API[3] in python. In order to compute statistical measurements of transaction's graphical patterns (i.e., fan-in, fan-out, scatter gathered), we considered only last 100 transactions from that address. For machine learning modeling, training, and testing, we used *Weka* library in java [4].

B. Accuracy

The accuracy of the designed model is tested using k-fold cross validation (where k is 10). With k-cross validation method, the whole dataset is split into k mutually exclusive parts. The model is trained and tested k time, where at each trial one part is considered as test data, whereas all the other parts are considered as training data. For each of k built models, the evaluation scores are retained, and final model is selected based on those scores with higher accuracy rate. In k-fold cross validation, each sample is used in the test set exactly once and to train the model, it is used $k-1$ times.

The correctness of the model is assessed using $accuracy$, $precision$, and $recall$ parameters. These parameters are computed based on 1) true positives (*TP*), which is the number of correctly identified mixers addresses, 2) true negatives (*TN*), which is the number of non-mixers addresses correctly identified as non-mixers, 3) false positives (*FP*), that is the number of non-mixers samples incorrectly detected as mixers, 4) false negatives (*FN*), which is the number of mixers addresses incorrectly detected as non-mixers. The $accuracy$, $precision$, and $recall$ are computed as Equation (2), (3), and (4), respectively.

$$Accuracy = (TP + TN)/(TP + TN + FP + FN) \quad (2)$$

$$Precision = (TP)/(TP + FP) \quad (3)$$

$$Recall = TP/(TP + FN) \quad (4)$$

The results for each of the correctness parameters are shown in Table III. On full dataset with 36 features, the C4.5-based tree achieved more than 97% *accuracy* and *precision*, and slightly less than 97% *recall*. If we reduced the tree with REP, the correctness is very slightly reduced, compared to a notable reduction in the size of the tree. Computing just 8 parameters against each address and using the tree size of just 55 nodes and 28 leaves, we achieved the accuracy of around 97%.

[3]https://api.blockcypher.com
[4]https://www.cs.waikato.ac.nz/ml/weka/

TABLE III
ACCURACY RESULTS

	TP	*FN*	*FP*	*TN*	*Accuracy*	*Precision*	*Recall*
Full dataset (without REP)	3098	101	72	3528	0.974	0.977	0.968
Full dataset (with REP)	3090	109	88	3513	0.971	0.972	0.966
Reduced dataset (without REP)	3089	110	80	3521	0.972	0.974	0.966
Reduced dataset(REP)	3075	124	94	3507	0.968	0.970	0.961

C. Efficiency

We are taking basic address-parameters–such as, transactions data, no. of input, no. of output, coin sent, received, etc.–from external server that holds the current state of the bitcoin blockchain. So, the time to extract these parameters from the blockchain depends on multi-factors, including the Internet speed and bandwidth, delay, and the servers computation power. In our case, we are taking data from *blockcypher.com* where it takes less than $200ms$ per address to retrieve basic parameters, however there is a lot of variations in this time due to above-mentioned factors. On the other hand, at our local machine, it takes less than $1ms$ to compute the selected 8 features from the basic ones. The decision-time taken by the model is itself very negligible, as the model has maximum of 55 conditional rules (with REP using top 8 address-features). With these results, and having blockchain node at our local machine, it is very feasible to detect mixers services in each of the coming blocks in the bitcoin network.

VI. CONCLUSION

Mixers detection is very essential for authorities to detect and cater money laundering services. In this paper, we proposed a decision-tree based mixer's detection model that is highly accurate and efficient. The model is developed using C4.5 decision tree learning approach with just eight features. For selecting the best eight features, *information gain* is computed against each of the features in the overall feature-set and the ones with higher *information gain* are selected for modeling. The developed model have the accuracy of more than 97% with a short tree of just 55 nodes.

ACKNOWLEDGMENT

The current work is supported by Atlantic Innovation Fund and MITACS (IT24468).

REFERENCES

[1] F. Reid and M. Harrigan, "An analysis of anonymity in the bitcoin system," in *Security and privacy in social networks*. Springer, 2013, pp. 197–223.

[2] E. Androulaki, G. O. Karame, M. Roeschlin, T. Scherer, and S. Capkun, "Evaluating user privacy in bitcoin," in *International conference on financial cryptography and data security*. Springer, 2013, pp. 34–51.

[3] S. Nakamoto and A. Bitcoin, "A peer-to-peer electronic cash system," *Bitcoin.–URL: https://bitcoin. org/bitcoin. pdf*, vol. 4, p. 2, 2008.

[4] D. Bradbury, "The problem with bitcoin," *Computer Fraud & Security*, vol. 2013, no. 11, pp. 5–8, 2013.

[5] M. Bartoletti, B. Pes, and S. Serusi, "Data mining for detecting bitcoin ponzi schemes," in *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*. IEEE, 2018, pp. 75–84.

[6] S. Foley, J. R. Karlsen, and T. J. Putniņš, "Sex, drugs, and bitcoin: How much illegal activity is financed through cryptocurrencies?" *The Review of Financial Studies*, vol. 32, no. 5, pp. 1798–1853, 2019.

[7] J. Pakki, Y. Shoshitaishvili, R. Wang, T. Bao, and A. Doupé, "Everything you ever wanted to know about bitcoin mixers (but were afraid to ask)," in *International Conference on Financial Cryptography and Data Security*. Springer, 2021, pp. 117–146.

[8] I. Alarab, S. Prakoonwit, and M. I. Nacer, "Comparative analysis using supervised learning methods for anti-money laundering in bitcoin," in *Proceedings of the 2020 5th International Conference on Machine Learning Technologies*, 2020, pp. 11–17.

[9] L. Cai and B. Wang, "Research on tracking and tracing bitcoin fund flows," in *2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC)*. IEEE, 2018, pp. 1495–1499.

[10] A. Gaihre, S. Pandey, and H. Liu, "Deanonymizing cryptocurrency with graph learning: The promises and challenges," in *2019 IEEE Conference on Communications and Network Security (CNS)*. IEEE, 2019, pp. 1–3.

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

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

[13] S. Ranshous, C. A. Joslyn, S. Kreyling, K. Nowak, N. F. Samatova, C. L. West, and S. Winters, "Exchange pattern mining in the bitcoin transaction directed hypergraph," in *International conference on financial cryptography and data security*. Springer, 2017, pp. 248–263.

[14] M. Jourdan, S. Blandin, L. Wynter, and P. Deshpande, "Characterizing entities in the bitcoin blockchain," in *2018 IEEE international conference on data mining workshops (ICDMW)*. IEEE, 2018, pp. 55–62.

[15] M. M. Rathore, S. A. Shah, D. Shukla, E. Bentafat, and S. Bakiras, "The role of ai, machine learning, and big data in digital twinning: A systematic literature review, challenges, and opportunities," *IEEE Access*, vol. 9, pp. 32 030–32 052, 2021.

[16] M. M. Rathore, A. Ahmad, and A. Paul, "Real time intrusion detection system for ultra-high-speed big data environments," *The Journal of Supercomputing*, vol. 72, no. 9, pp. 3489–3510, 2016.

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

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

[19] J. R. Quinlan, *C4. 5: programs for machine learning*. Elsevier, 2014.

[20] T. d. Balthasar and J. Hernandez-Castro, "An analysis of bitcoin laundry services," in *Nordic Conference on Secure IT Systems*. Springer, 2017, pp. 297–312.

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

[22] M. M. U. Rathore, M. J. J. Gul, A. Paul, A. A. Khan, R. W. Ahmad, J. J. Rodrigues, and S. Bakiras, "Multilevel graph-based decision making in big scholarly data: An approach to identify expert reviewer, finding quality impact factor, ranking journals and researchers," *IEEE Transactions on Emerging Topics in Computing*, vol. 9, no. 1, pp. 280–292, 2018.

[23] M. M. Rathore, S. Attique Shah, A. Awad, D. Shukla, S. Vimal, and A. Paul, "A cyber-physical system and graph-based approach for transportation management in smart cities," *Sustainability*, vol. 13, no. 14, p. 7606, 2021.

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

[25] L. Novetta, "Survey of bitcoin mixing services: Tracing anonymous bitcoins," *White Paper*, 2015.

[26] A. Shojaeenasab, A. P. Motamed, and B. Bahrak, "Mixing detection on bitcoin transactions using statistical patterns," *arXiv preprint arXiv:2204.02019*, 2022.
[27] H. S. Yin and R. 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)*. IEEE, 2017, pp. 3690–3699.
[28] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.
[29] M. A. Harlev, H. Sun Yin, K. C. Langenheldt, R. Mukkamala, and R. Vatrapu, "Breaking bad: De-anonymising entity types on the bitcoin blockchain using supervised machine learning," in *Proceedings of the 51st Hawaii international conference on system sciences*, 2018.
[30] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," *Advances in neural information processing systems*, vol. 30, 2017.
[31] S. Al-Farsi, M. M. Rathore, and S. Bakiras, "Security of blockchain-based supply chain management systems: challenges and opportunities," *Applied Sciences*, vol. 11, no. 12, p. 5585, 2021.
[32] J. R. Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986.
[33] K. Toyoda, T. Ohtsuki, and P. T. Mathiopoulos, "Multi-class bitcoin-enabled service identification based on transaction history summarization," in *2018 IEEE international conference on internet of things (iThings) and IEEE green computing and communications (GreenCom) and IEEE cyber, physical and social computing (CPSCom) and IEEE smart data (SmartData)*. IEEE, 2018, pp. 1153–1160.