ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 12, December 2017

Anomaly Detection in Block chain Using Clustering Protocol

[1] Rachana Kumari, [2] Monica Catherine
[1] M.Tech (ISCF), SRM University, Kattanakulathur, India
[2] Assistant Professor, IT Department, SRM University, Kattanakulathur, India

***Abstract*: With the growth of technology, everyday new technologies are evolving. Blockchain is one such technology. Blockchain is a revolutionary technology which has proved its potential of being used in numerous fields like digital money or crypto-currency, IoT, Product tracing, Smart Contracts etc. Blockchain is a distributed database which allows to share or process data between multiple parties over a network of non-trusted users securely. One of the biggest advantages of blockchain is that it is fully decentralized i.e. there is no central authority which is governing it. However, it has its own set of disadvantages also. Some of these problems are its complexity, network size, large energy consumption etc. But one key problem is that there is no way to find the anomalous node i.e. nodes which are malicious. In any Blockchain network, there are two types of nodes, the one which behaves normally i.e. the honest node. But some nodes may try to cheat in the network or may have some illegal interest I.e. malicious nodes. If someone tries to monitor this node behaviour manually, it will take tremendous time and effort and is nearly impossible.**

So, this paper introduces a novel solution for the above mentioned problem. The problem can be solved by clustering the nodes of the network. For this we will propose an algorithm which will help us in clustering the blockchain and then further for analysing the malicious activity of the nodes, if any performed. In this we will divide the whole network into clusters of nodes or data points depending or based on some similar traits that they may have. Therefore the aim of our project is to segregate groups having similar traits from the blockchain network and then cluster them so that to identify the malicious node or illegal behaviour.

***Key Words:* Blockchain, Cluster, Clustering Algorithm, K-means clustering, Security.**

I. INTRODUCTION

A blockchain is a scattered database that keeps a list of growing records of all the communications occurred. The growing records are termed blocks. All block holds a timestamp, a nonce, a mention to (ie. hash of) the former block and a list of all of the transactions that have taken place since the preceding block. Unlike from old-fashioned central records, blockchain does not have a centralized node. Every node in the blockchain linkage has an equivalent right (in query, in directing transactions, also in participating into the consensus process) and keeps up a ledger that registers every single transaction that has ever happened.

For a public blockchain, everyone in the world can connect to the blockchain network and become a node. For a private chain, only the contributing parties that are certified or have a proper authorization can become nodes.

Since every node in the blockchain network preserves a ledger of all the transactions happened, there are multi-backup ledgers in the blockchain network. As such, the openness of a centralized database is eradicated. Additionally, since every blocks in the blockchain network cannot be reformed or adjusted, blockchain is safeguarded from data alteration and false modification.

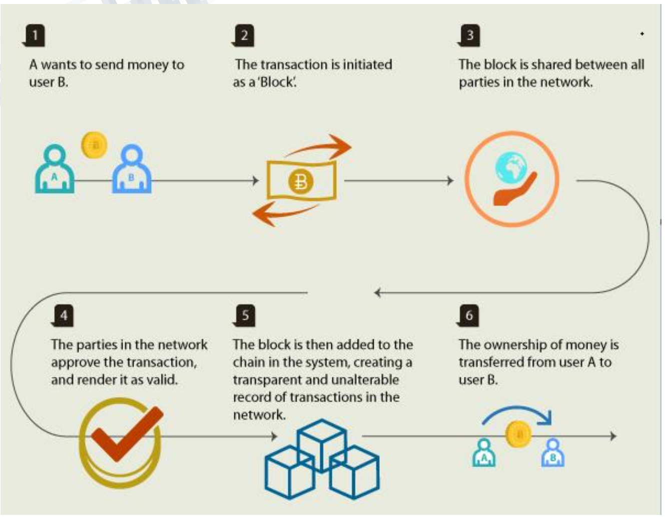

1
A wants to send money to user B.
A
B
2
The transaction is initiated as a 'Block'.
3
The block is shared between all parties in the network.
4
The parties in the network approve the transaction, and render it as valid.
5
The block is then added to the chain in the system, creating a transparent and unalterable record of transactions in the network.
6
The ownership of money is transferred from user A to user B.
A
B

Fig: Working of Blockchain

Blockchain has a potential and ability for being the ground-breaking technology that has the prospective to discover various applications in numerous fields. BitCoin has revealed how blockchain technologies can facilitate the making of a crypto-currency. In more modern periods,

blockchain applications have seemed to drive future outside their paramount application -BitCoin [5, 8]. For instance, blockchain technology has found its applications in fields as varied as transaction processing, government cash management, clearing, medicine tracing, gambling, and personal credit management system [1,3].
However, blockchain faces its own problems and challenges. In a public blockchain, there are usually millions of nodes. Some nodes may attempt to cheat in the network for illegal interests and have anomalous behavior patterns while the majority nodes behaves normally.

According in conformity with the overall performance concerning nodes between blockchain transactions, nodes can remain divided in helpful nodes, and malicious nodes.

1. Helpful nodes. When offering work according to mean nodes, nodes along a higher behavioural toughness then popularity value, who execute ascertain the rightness on information forwarded after lousy nodes, are known so helpful nodes.

2. Malicious nodes. Nodes juggle or throw abroad data, then even worms and Trojans are forged so facts to remain forwarded to others. These conduct regarding nodes execute severely harm the security or toughnessconcerning blockchain communication. Nodes as celebrate disguised data and demolish resources are acknowledged namely malicious nodes. It takes a large amount of time and efforts, if possible, according to manually discipline the behaviors regarding all the nodes. Towards that issue, this paper propose to routinely cluster the behavior patterns concerning all the nodes between categories. After the clustering, we may pick out consultant conduct patterns because of every category as behavior templates. Then utilizes the conduct templates after pick out curious behavior patterns up to expectation work now not conform according to any template. Moreover, clustering conduct patterns in categories may each leads to deeper insights among the blockchain network then helps maintainers rule then arrange the nodes.This bill seeks in imitation of automatically fascicle the behavior patterns into categories.

The principal contributions about it demand bill be able remain short as like below.

- We eliminate sequences facts according to the node behaviors, or advise an algorithm in conformity with brush nodes within categories.

To the superior concerning our knowledge, this demand bill is the first according to formulate and address the problem regarding clustering node behaviors between blockchain networks.

- We propulsize huge experiments in conformity which evaluate the usefulness concerning our algorithm towards the present techniques and instruction its workings within a variety of settings. Experimental results wish exhibit that our proposed algorithm is plenty greater superb than the current methods of terms on clustering accuracy.

II. RELATED WORK

In order to cluster the behavior patterns of all the nodes in blockchain network, the first step is to extract features to represent behavior pattern of each node. Since the most important feature of a node is its transaction amount changing over time, we extract the sequences according to the transaction amounts change over time. Each node is represented by a sequence. Our goal is to cluster these sequences into several categories where the number of categories is defined by the user. The most closely related work are sequence similarity measure and clustering approaches.

A. Sequence similarity measure

Sequence similarity measure defines the similarity between two sequences being compared. Numerous research in sequences has produced a number of distance measures [2]. The most popular measures are Euclidean distance, Dynamic Time Warping (DTW), Edit Distance on Real sequence (EDR) and Longest Common SubSequences (LCSS). Euclidean distance requires the two sequences be of the same length. It computes the distance by simply employing the L2 norm. DTW does not require that the two sequences be of the same length and can handle time shifting. To implement this, DTW duplicates the previous elements and calculates an optimal match between the sequences. The LCSS technique [10] introduces a threshold value _ to handle noises in sequences, which try to find the longest common subsequences between the given two sequences. EDR [2] leverages gap and mismatch penalties and can handle both noises and time shifting.

B. Clustering approaches

Numerous clustering techniques have been proposed within the last decades, who may keep at large labelled into four main categories: partitioning-based methods, density-based methods, hierarchical methods, or grid-based methods [3, 5, 6, 7].

Partitioning-based strategies percentage the n unlabelled statistics tuples in ok partitions. Each percentage corresponds according to a fascicle and has a cluster center. Two renowned heuristic strategies according to symbolize division primarily based strategies are k-means or k-medoids. In k-means, every fascicle is represented via the average worth about the tuples into the cluster. In contrast, in k-medoids, every tussock is represented by way of the just centrally located tuple into a cluster.

Hierarchical strategies crew data tuples within a grower shape [9]. There are typically twain sorts over hierarchical methods: agglomerative then divisive.

- Agglomerative techniques start by concerning every tuple as a fascicle andsince submerse clusters in larger and large clusters, till entire tuples are among a singular lot then the desired quantity on clusters are satisfied.

- Divisive strategies assignment of the contrary way, it places all tuples within the equal cluster yet since division the clusters between smaller or smaller clusters. A primary hassle within hierarchical clustering techniques is up to expectation any immerse or split is unchanging as soon as executed. Chameleon[10] yet BIRCH[16] are couple traditional hierarchical methods. Density based totally [4] strategies starts beyond a tussock of certain tuple yet hold absorbing neighbour tuples as much lengthy namely the closeness (number over tuples inside a assured distance) is higher than a threshold. DBSCAN is the just classic technique in density based clustering. These methods, however, require to pre-set a not much parameters. It is commonly challenging to pick out a excellent parameter values. Grid-based methods [1, 12] quantize the function house of a finite variety regarding cells such grid structure. A regular instance concerning the grid-based tactics is STING. In STING, numerous ranges regarding quadrate cells correspond according to quite a few extraordinary ranges over resolution. Statistical records for every phone is pre-computed then stored. Due in conformity with the grid structure, it strategies hold fast computation speeds, the worth is losing a section over accuracy.

III. PROPOSED SOLUTION

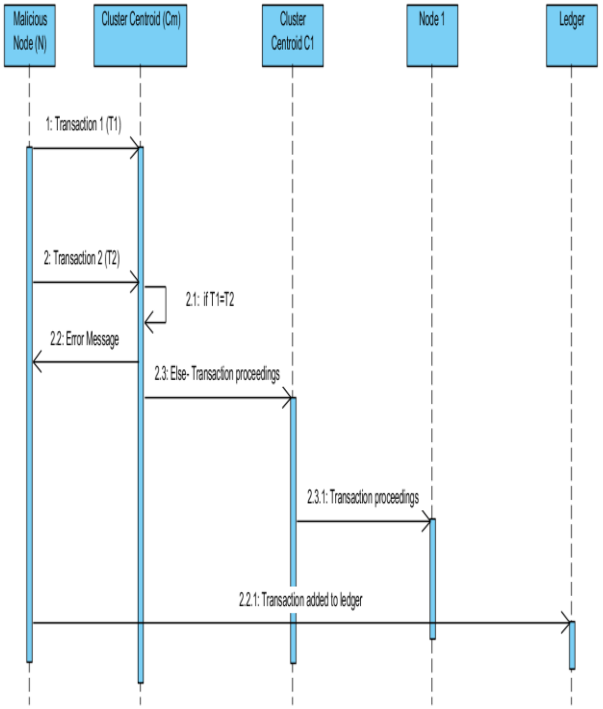
We know that even though blockchain technology can prevent fraud and anonomyous behaviour, it cannot detect fraud by its own. Attacker can find one way or the other to steal currency and can commit various fraudulent crime. Also with the increase in number of attacks or vulnerabilities, the developers are trying to come up with new and innovative technique and technologies to strengthen the blockchain and also to detect the attack prior to its occurrence.

Techniques such as machine learning and data mining algorithms can be very much helpful in finding and detecting malicious behaviour, fraud and intrutions in the blockchain based transactions. By profiling, monitoring, and detecting behavioural patterns based on people's transaction histories and transaction time, supervised machine learning approaches, such as deep-learning neural networks, support vector machines, and Bayesian belief networks may help detect outlier behaviours. So,we also will use one amongs these technologies to provide security agains fraud and malicious behaviour.

Here, in this paper we propose a system in which the blockchain will be clusterd in groups on the basis of its behavioural pattern. To analyse the behaviour, the parameter that we are choose as our pattern is the transaction amount change over time for the nodes. The reason for selecting this parameter is that usually the transaction amount is the most important and predominant feauture of a node. For doing this the algorithm that will be used is K-means algorithm. Also we will not use the same algorithm but a little modified version of this algorithm.

A. Sequence Diagram

The diagram shows how the clusters objects interact with each other. Through this diagram we can understand how the clustering will help in securing the blockchain. Let us assume that there is a malicious node (M) and its cluster centroid is Cm. If M initiates two transaction say T1 and T2 at the same time (attack called double spending attack) to two different node, then the Cm will get to know about it as the transaction will be added to the block. When Cm gets to know that some kind of malicious activity is being performed by the node M, it will discard both transactions and show an error message.



If the transaction are to two genuine users and the node is not doing any malicious activity, then the transactions will be sent from the Cm to the Cluster centroids of the other node i.e the node which has to accept the transaction. And that centroid will proceed the transaction to the node. After all this the transaction whether valid or invalid will be added to the distributed ledger, which keeps record of all the transaction happened ever.

B. Methodology

In this section, we introduce our method to address the problem defined in the paper. In order to cluster the sequences into clusters, the first step is to select a similarity measure for comparing two sequences. We first present an introduction to the similarity measures in Section 1 and then elaborate our clustering algorithm termed K-means Clustering algorithm, which we will improvise to create our own algorithm, in Section 2.

1. Similarity Measure Selection

Sequences similarity measure defines the similarity between two sequences being compared. Numerous sequence similarity measures have been proposed namely Euclidean distance, Dynamic Time Warping (DTW), Edit Distance on Real sequence (EDR) and Longest Common SubSequences (LCSS).

Since DTW distance is commonly adopted and can deal with two sequences with different lengths, we select DTW distance as the similarity measure between sequences. Note that similarity is inversely proportional to distance. EDR distance and LCCS similarity can also handle two sequences of different lengths.We do not select them as the similarity measure for the following reasons. EDR distance and LCCS similarity are mainly designed to handle the noises in the sequences. However, all the transaction amounts in the blockchain network are precise and no noise is allowed.

2. K-Means Clustering Algorithm

K-means clustering is a method by which elements of dataset are divided into K different groups based on similarity to one another.

Current Algorithm Summary:

1. Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids.

2. Assign each object to the group that has the closest centroid.

3. When all objects have been assigned, recalculate the positions of the K centroids.

4. Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.

C. Proposed Improvisation:

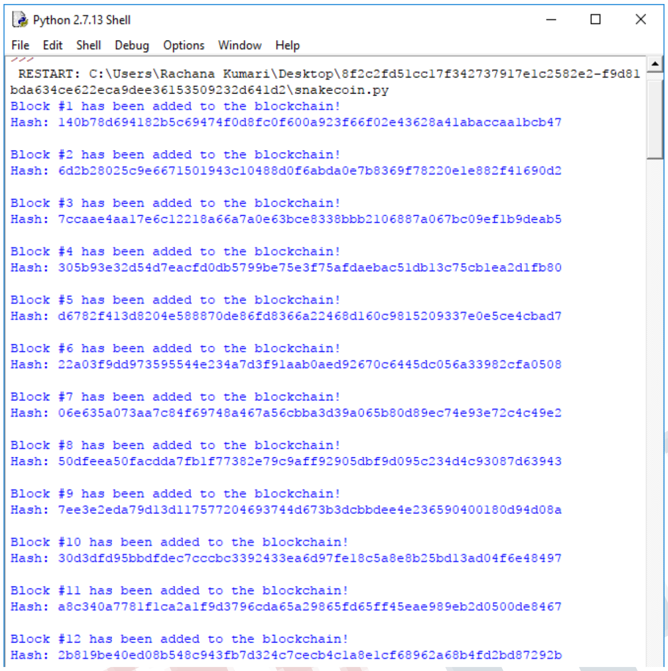
• K-means clustering initializes the cluster centers randomly, while we want our algorithm to sort the sequences and select k sequences uniformly from the sorted list.

• k-means clustering utilizes Euclidean distance between static tuples, while our algorithm will utilizes DTW distance between sequences.

• K-means clustering uses the average value of the tuples in the cluster as the cluster center for that cluster, whileour algorithm will selects the sequence with the smallest distance to its [n/k]th nearest neighbor among all sequences in the cluster as the cluster center.

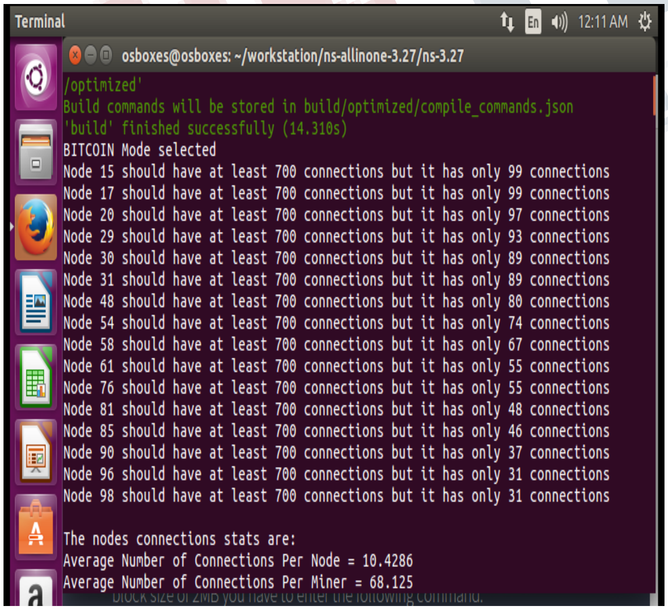
IV. RESULT

OUTPUT (BLOCKCHAIN CREATION)



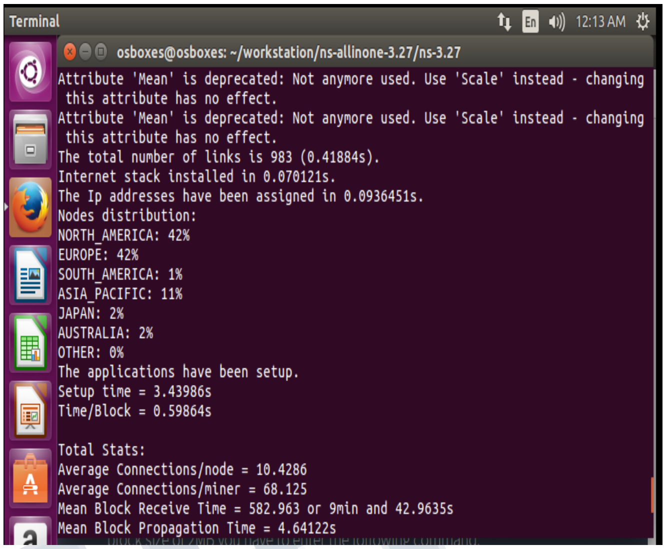
```
 RESTART: C:\Users\Rachana Kumari\Desktop\8f2c2fd51cc17f342737917e1c2582e2-f9d81
bda634ce622eca9dee36153509232d641d2\snakecoin.py
Block #1 has been added to the blockchain!
Hash: 140b78d694182b5c69474f0d8fc0f600a923f66f02e43628a41abaccaa1bcb47

Block #2 has been added to the blockchain!
Hash: 6d2b28025c9e6671501943c10488d0f6abda0e7b8369f78220e1e882f41690d2

Block #3 has been added to the blockchain!
Hash: 7ccaae4aa17e6c12218a66a7a0e63bce8338bbb2106887a067bc09ef1b9deab5

Block #4 has been added to the blockchain!
Hash: 305b93e32d54d7eacfd0db5799be75e3f75afdaebac51db13c75cb1ea2d1fb80

Block #5 has been added to the blockchain!
Hash: d6782f413d8204e588870de86fd8366a22468d160c9815209337e0e5ce4cbad7

Block #6 has been added to the blockchain!
Hash: 22a03f9dd973595544e234a7d3f91aab0aed92670c6445dc056a33982cfa0508

Block #7 has been added to the blockchain!
Hash: 06e635a073aa7c84f69748a467a56cbba3d39a065b80d89ec74e93e72c4c49e2

Block #8 has been added to the blockchain!
Hash: 50dfeea50facdda7fb1f77382e79c9aff92905dbf9d095c234d4c93087d63943

Block #9 has been added to the blockchain!
Hash: 7ee3e2eda79d13d117577204693744d673b3dcbbdee4e236590400180d94d08a

Block #10 has been added to the blockchain!
Hash: 30d3dfd95bbdfdec7cccbc3392433ea6d97fe18c5a8e8b25bd13ad04f6e48497

Block #11 has been added to the blockchain!
Hash: a8c340a7781f1ca2a1f9d3796cda65a29865fd65ff45eae989eb2d0500de8467

Block #12 has been added to the blockchain!
Hash: 2b819be40ed08b548c943fb7d324c7cecb4c1a8e1cf68962a68b4fd2bd87292b
```

Fig: Output Block Created



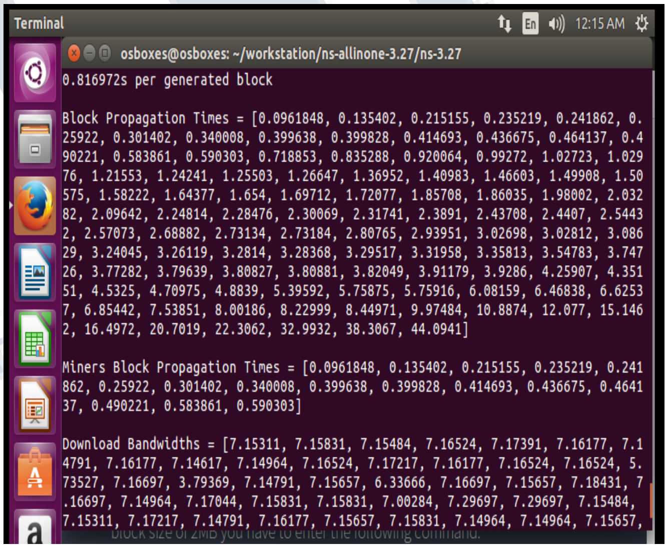
```
/optimized'
Build commands will be stored in build/optimized/compile_commands.json
'build' finished successfully (14.310s)
BITCOIN Mode selected
Node 15 should have at least 700 connections but it has only 99 connections
Node 17 should have at least 700 connections but it has only 99 connections
Node 20 should have at least 700 connections but it has only 97 connections
Node 29 should have at least 700 connections but it has only 93 connections
Node 30 should have at least 700 connections but it has only 89 connections
Node 31 should have at least 700 connections but it has only 89 connections
Node 48 should have at least 700 connections but it has only 80 connections
Node 54 should have at least 700 connections but it has only 74 connections
Node 58 should have at least 700 connections but it has only 67 connections
Node 61 should have at least 700 connections but it has only 55 connections
Node 76 should have at least 700 connections but it has only 55 connections
Node 81 should have at least 700 connections but it has only 48 connections
Node 85 should have at least 700 connections but it has only 46 connections
Node 90 should have at least 700 connections but it has only 37 connections
Node 96 should have at least 700 connections but it has only 31 connections
Node 98 should have at least 700 connections but it has only 31 connections

The nodes connections stats are:
Average Number of Connections Per Node = 10.4286
Average Number of Connections Per Miner = 68.125
```

Fig: Node and Connection Creation



```
Attribute 'Mean' is deprecated: Not anymore used. Use 'Scale' instead - changing
 this attribute has no effect.
Attribute 'Mean' is deprecated: Not anymore used. Use 'Scale' instead - changing
 this attribute has no effect.
The total number of links is 983 (0.41884s).
Internet stack installed in 0.070121s.
The Ip addresses have been assigned in 0.0936451s.
Nodes distribution:
NORTH_AMERICA: 42%
EUROPE: 42%
SOUTH_AMERICA: 1%
ASIA_PACIFIC: 11%
JAPAN: 2%
AUSTRALIA: 2%
OTHER: 0%
The applications have been setup.
Setup time = 3.43986s
Time/Block = 0.59864s

Total Stats:
Average Connections/node = 10.4286
Average Connections/miner = 68.125
Mean Block Receive Time = 582.963 or 9min and 42.9635s
Mean Block Propagation Time = 4.64122s
```

Fig: Node Deployment



```
0.816972s per generated block

Block Propagation Times = [0.0961848, 0.135402, 0.215155, 0.235219, 0.241862, 0.
25922, 0.301402, 0.340008, 0.399638, 0.399828, 0.414693, 0.436675, 0.464137, 0.4
90221, 0.583861, 0.590303, 0.718853, 0.835288, 0.920064, 0.99272, 1.02723, 1.029
76, 1.21553, 1.24241, 1.25503, 1.26647, 1.36952, 1.40983, 1.46603, 1.49908, 1.50
575, 1.58222, 1.64377, 1.654, 1.69712, 1.72077, 1.85708, 1.86035, 1.98002, 2.032
82, 2.09642, 2.24814, 2.28476, 2.30069, 2.31741, 2.3891, 2.43708, 2.4407, 2.5443
2, 2.57073, 2.68882, 2.73134, 2.73184, 2.80765, 2.93951, 3.02698, 3.02812, 3.086
29, 3.24045, 3.26119, 3.2814, 3.28368, 3.29517, 3.31958, 3.35813, 3.54783, 3.747
26, 3.77282, 3.79639, 3.80827, 3.80881, 3.82049, 3.91179, 3.9286, 4.25907, 4.351
51, 4.5325, 4.70975, 4.8839, 5.39592, 5.75875, 5.75916, 6.08159, 6.46838, 6.6253
7, 6.85442, 7.53851, 8.00186, 8.22999, 8.44971, 9.97484, 10.8874, 12.077, 15.146
2, 16.4972, 20.7019, 22.3062, 32.9932, 38.3067, 44.0941]

Miners Block Propagation Times = [0.0961848, 0.135402, 0.215155, 0.235219, 0.241
862, 0.25922, 0.301402, 0.340008, 0.399638, 0.399828, 0.414693, 0.436675, 0.4641
37, 0.490221, 0.583861, 0.590303]

Download Bandwidths = [7.15311, 7.15831, 7.15484, 7.16524, 7.17391, 7.16177, 7.1
4791, 7.16177, 7.14617, 7.14964, 7.16524, 7.17217, 7.16177, 7.16524, 7.16524, 5.
73527, 7.16697, 3.79369, 7.14791, 7.15657, 6.33666, 7.16697, 7.15657, 7.18431, 7
.16697, 7.14964, 7.17044, 7.15831, 7.15831, 7.00284, 7.29697, 7.29697, 7.15484,
7.15311, 7.17217, 7.14791, 7.16177, 7.15657, 7.15831, 7.14964, 7.14964, 7.15657,
```

Fig: Block and Miner Propogation

CONCLUSION

In this paper, we present the problem of behaviour pattern clustering in blockchain networks and have proposed a novel algorithm to address this problem. To the best of our knowledge, this paper is the first to formulate and address the problem of clustering node behaviors in blockchain networks. We will evaluate a list of potential sequence similarity measures, and select a distance that is suitable for the behavior pattern clustering problem.

REFERENCES

1. Christidis K, Devetsikiotis M (2016) Blockchains and smart contracts for the internet of things. IEEE Access 4:2292–2303

2. Croman K, Decker C, Eyal I, Gencer AE, Juels A, Kosba AE, Miller A, Saxena P, Shi E, Sirer EG, Song D, Wattenhofer R (2016) On scaling decentralized blockchains - (a position paper). In: Financial cryptography and data security - FC 2016 international workshops, BITCOIN, VOTING, and WAHC, pp 106–125

3. Dorri A, Kanhere SS, Jurdak R (2016) Blockchain in internet of things: challenges and solutions. arXiv:1608.05187

4. Ester M, Kriegel H, Sander J, Xu X (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the second international conference on knowledge discovery and data mining (KDD-96), pp 226–231

5. Garay JA (2015) Blockchain-based consensus (keynote). In: 19th international conference on principles of distributed systems, OPODIS 2015, pp 5:1–5:1

6. Guadamuz A, Marsden C (2015) Blockchains and bitcoin: regulatory responses to cryptocurrencies. First Monday 20(12)

7. Guha S, Rastogi R, Shim K (2001) Cure: an efficient clustering algorithm for large databases. Inf Syst 26(1):35–58

8. Hirano S, Tsumoto S (2003) Comparison of similarity measures and clustering methods for timeseries medical data mining. In: Data mining and knowledge discovery: theory, tools, and technology, pp 219–225

9. Karame G (2016) On the security and scalability of bitcoin’s blockchain. In: Proceedings of the 2016 ACM SIGSAC conference on computer and communications security, Vienna, Austria, October 24–28, 2016, pp 1861–1862

10. Karypis G, Han E, Kumar V (1999) Chameleon: hierarchical clustering using dynamic modeling. IEEE Computer 32(8):68–75

11. Liao TW (2005) Clustering of time series data: a survey. Pattern Recogn 38(11):1857–1874

12. Morse MD, Patel JM (2007) An efficient and accurate method for evaluating time series similarity. In: Proceedings of the ACM SIGMOD international conference on management of data, Beijing, China, June 12–14, 2007, pp 569–580

13. Underwood S (2016) Blockchain beyond bitcoin. Commun ACM 59(11):15–17

14. Xu R, Wunsch II DC (2005) Survey of clustering algorithms. IEEE Trans Neural Networks 16(3):645–678

15. Zhang T, Ramakrishnan R, Livny M (1996) BIRCH: An efficient data clustering method for very large databases. In: Proceedings of the 1996 ACM SIGMOD international conference on management of data, pp 103–114

16. Zyskind G, Nathan O, Pentland A (2015) Decentralizing privacy: using blockchain to protect personal data. In: 2015 IEEE symposium on security and privacy workshops, SPW 2015, pp v180–184