TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Ashu Dhama 184635IASM

Secured decentralized IoT based on blockchain based technology

Master's thesis

Supervisor: Muhidul Islam Khan

Tallinn 2020

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Ashu Dhama 184635IASM

Turvaline detsentraliseeritud IoT, mis põhineb plokiahelal põhineval tehnoloogial

Magistritöö

Juhendaja: Muhidul Islam Khan

Tallinn 2020

Author’s declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Ashu Dhama

13.12.2020

Abstract

Internet of things (IoT) consist of millions of devices producing data on a massive scale. IoT is a complex network that contains various physical objects such as sensors, software, actuators and other different technologies which share data among each other over the internet. Moreover, there are variety of major issues during this exceptional growth of IoT, especially in the field of privacy and security. Blockchain can be used as a ledger to store the data for IoT applications to provide high level of privacy and security to the IoT applications. However, blockchain has many challenges with IoT those need to be resolved where scalability and security are the major challenges. In context of IoT blockchain needed high computational power to solve the proof of work, to improve scalability and low latency for transactions confirmation in the network. This thesis proposed a framework for secured IoT where I considered miner selection with three important properties such computing power, block propagation delay and hash generation rate. Learning to rank algorithm (LTR) is applied in order to get the best node in terms of hash generation rate, block propagation delay and computing power as a miner. Miner node is used to create signature for each new block in the blockchain by using hashing technique. LTR performs outstanding over the random miner selection as it scored all nodes based on a scoring function, the node with best properties has the highest scoring value among all nodes in the network. Finally, I compare the proposed method with existing random miner selection procedure. Results from the simulation shows that proposed method improves blockchain scalability with IoT because its choses the miner with best properties mainly in terms of computing power, hash generation rate and block propagation dely.

Annotatsioon

Asjade Internet (Internet of Things - IoT) koosneb millionitest seadmetest, mis toodavad andmeid massiivsel skaalal. IoT on kompleksne võrk, mis hõlmab erinevaid füüsilisi komponente nagu sensorid ja ajamid, tarkvara ja muid tehnoloogiaid, mis jagavad andmeid omavahel läbi Interneti. Lisaks kaasneb Asjade Interneti eksponentsiaalse kasvuga mitmeid probleeme, eriti privaatsuse ja turvalisuse vallas. IoT rakenduste andmete hoiustamiseks saab kasutada Plokiahelat (Blockchain), et tagada kõrgel tasemel privaatsus ja turvalisus. Plokiahelel on siiski IoT-ga palju väljakutseid, mida lahendada, millest olulisemad on mastaapsus ja turvalisus. IoT kontekstis vajas plokiahel suurt arvutusvõimsust, et lahendada Proof of Work algoritm, parandada mastaapsust ja tagada madal latentsusaeg tehingute kinnitamisel võrgus. Selles lõputöös pakuti välja turvalise IoT raamistik, kus ma kaalusin kaevurite valimist kolme olulise omadusega - arvutusvõimsus, ploki levimise viivitus ja räsi genereerimise kiirus. Learning to Rank (LTR) algoritmi kasutatakse selleks, et saada parim kaevur nende kolme omadusega. Kaevuri sõlme kasutatakse plokiahela iga uue ploki allkirja loomiseks räsimistehnika abil. LTR toimib juhusliku kaevurivaliku osas silmapaistvalt, kuna skooris kõik sõlmed hindamisfunktsiooni põhjal. Parimate omadustega sõlmel on kõigi võrgu sõlmede seas suurim punktisumma. Lõpuks võrdlen pakutud meetodit olemasoleva juhusliku kaevurite valimise protseduuriga. Simulatsiooni tulemused näitavad, et pakutud meetod parandab plokiahela mastaapsust IoT-ga, kuna see valib parimate omadustega kaevuri peamiselt arvutusvõimsuse, räsi genereerimise kiiruse ja blokeerimise leviku viivituse osas.

List of abbreviations and terms

ML	Machine Learning
RMS	Random Miner Selection
LTR	Learning to Rank
TX	Transaction
DoS	Denial of Service
IoT	Internet of Things
CP	Computing Power
hGr	Hash Generation Rate
Bpd	Block Propagation Delay
MDP	Markov Decision Process
Bitcoin NG	Bitcoin Next Generation

Table of contents

List of figures

List of tables

1 Introduction

Internet of Things (IoT) consist of millions of distributed devices which produce data on a massive scale. IoT is a complete network that consist of several physical objects for instance sensors, software and other different technologies that exchange and share data between each other over the internet. IoT is confluence of multiple technologies such as machine learning, embedded systems, real-time analytics, control systems, automation (including smart city, smart home), healthcare devices, security systems and so on [1]. In big smart cities, IoT data collected from many different services for example security services, health related services and transport services. Smart cities provide facility of real time monitoring of environmental management, security, personal care, energy management and different other services.

There are varieties of major issues during this exceptional growth of IoT, especially in the field of privacy and security. In traditional centralized IoT monitoring system, IoT data is generally sent to the cloud-based servers for further processing. However, it increases the risk of entire system breakdown if server providers fail due to it inherits central architecture and subsequently it contains risk of cyber attacks where denial of service (DoS) and Ransom are most common attacks. Sometimes cloud server can be inaccessible because of software or other maintenance issues. Cloud does not support an open source code behaviour which creates a feeling of doubt or suspicion among different consumers and vendors [2] .

A decentralized approach in peer to peer network, helps to improve reliability in IoT. However, security is a serious concern in case of decentralized IoT [3]. Recently blockchain appeared as a trusted method in peer to peer communication which removes the risk of attacks from hackers because it provides less entry points and high protection for system [4].

1.1 Background

Blockchain is a system of storing information in such a manner that hacking or cheating the system would be tough or impossible. It was invented by the person or group of people called by name Santoshi Nakamoto in 2008 to serve as a public ledger of cryptocurrency bitcoin [5].

Blockchain is capable to handle enormous amount of IoT data coming from huge number of connected devices in big smart cities. Blockchain is a distributed way of providing security, which is auditable, immutable and fault resistant. Blockchain provides a decentralized architecture storage for IoT technology however, it requires high computational power to handle a huge number of transactions.

In current technology world, blockchain is highly reliable and secure, while decentralization, transparency and immutability are three main pillars behind the blockchain technology success. Blockchain can act as a decentralized architecture to record the data and can be regarded as a public ledger. All transactions executed by blockchain are stored in a list of blocks where each block consists of the block header, transaction counter, and transaction data. The process of adding new block to the blockchain is called mining. Nodes involved in this mining process called as minors. Moreover, blockchain provides extra security using cryptographic and consensus algorithms where one block is added over another block by using hashing and some consensus algorithm for example proof of work. These cryptographic and consensus algorithms computed by miner node, so miner node required more computing power as compare to the non-miner node [6].

Blockchain is basically a chain of records that used to store information/data. These records also called as block where each block is connected to next and previous block with the help of cryptographic hash. Each block stores hash of previous block, timestamp and transaction data which is generally exist in the form of Merkle tree. Blockchain can be denoted as a distributed ledger where each node in the network follow a specific rule for internal communication and operations. We cannot easily tamper the blockchain database [7]. Hence, it eliminates the threat of alteration with data because each node contains same revision of data and it is impossible to update all nodes in the network because it needs very high level of computational power to compute cryptographic and consensus algorithms for all nodes. As per the growing network of IoT devices we need a secure method to handle this communication and blockchain is capable to handle this huge amount of IoT data [8].

1.2 Problem Description

Blockchain is capable to handle and process big amount of IoT data. Moreover, there are still challenges with blockchain technology which need to be resolved. The overall internet infrastructure needs to be increased to handle this huge data which is processed by large scale IoT systems. Although blockchain can provide decentralized peer to peer network, distributed file sharing and autonomous device coordination and allowing IoT systems to track the big number of connected network devices while it needs huge amount of processing power that can reduce the long delay between hash generation for each transaction. Even though blockchain can resolve the privacy and reliability related issues with IoT but it has some limitations those make it a challenge to use blockchain technology with IoT. The main challenge with blockchain technology for IoT is variations in processing speed and time. In the blockchain, each node is capable to get a copy of complete blockchain's information after joining the blockchain network. In blockchain network, there are different types of nodes or records for instance miner nodes – these are the most powerful nodes, general nodes - this are the nodes those are participating in mining activities and beginning node is the first node of the chain [9]. Miner nodes are most powerful nodes specially in terms of CPU processing and memory. Miner nodes used to generate target hash for each new block by consuming its own resources and do enough proof of work for each new block. In the end, miner node broadcast the block in blockchain network. This complete hash generation and consensus process consumes huge processing power because each node competes to be a miner. Efficient miner selection process needs to be developed in order to improve the mining process. Blockchain is a new technology and there are still many issues those are needed to be resolved to make it more convenient. As we know, IoT systems are diverse systems and connected over a vast network. Using blockchain with IoT become more complex when encryption algorithms need to be run for each transaction for multiple devices because it will consume more computing power. Typical random miner selection is not useful for blockchain technology with IoT because it does not evaluate miner based on any attributes

[10]. Hence, proper miner selection is a critical issue in the blockchain technology and we need an adaptive method to select the miner node on a particular time stamp.

1.3 Challenges

Hash generation and consensus for each new block takes more time with randomly selected miner node. This time can be too long because miner node is not verified based on its properties. Hence, random miner selection is not good enough for blockchain technology. Adaptive miner selection helps to get a miner with best capabilities for better scalability of blockchain with IoT [11]. However, getting best miner is difficult while all nodes competing to be a miner, and therefore the challenge of this thesis is to consider the adaptive miner selection in the proposed framework. The wide range of options to configure the blockchain raise the main challenge to adopt blockchain based technologies for securing IoT. However, scalability is a major challenge with blockchain. If we consider the blockchain in context of IoT then it has several other problems such as demands of high computational power to solve consensus, less space and long latency for transactions confirmation from the other nodes in the network [12].

Regularly increasing transactions are making blockchain bulky because block size is limited [1]. Miner need to do a lot of work for each block, sometimes transactions are delayed because miner is busy with other transactions. Although blockchain has a great potential and it can be applied with wide range of IoT applications in the field of engineering, but exact impact for each different field and technology must be studied clearly. Blockchain provides unbreakable security and flexibility in accessing data for IoT applications.

1.4 Goals

The main goal of the thesis is to propose a framework for secured IoT based on blockchain based technology that optimize the overall blockchain performance for the decentralized IoT. This framework is focused on machine learning based approach to select the best miner from all available nodes in the network.

The objective is to propose a decentralized mechanism using blockchain based technology for secure IoT. Blockchain technology can provide data privacy and high-level security for IoT based applications but scalability is a major challenge with blockchain. As this thesis is considering blockchain in context of IoT so main goal is to improve the blockchain scalability for IoT based system [13].

Moreover, the proposed framework is more focused on using machine learning based algorithm to provide a better approach to blockchain for selecting miner nodes.

1.5 Structure of thesis

In chapter 1, the thesis describes about IoT based applications and issues related to data security and privacy with IoT systems that can be resolved with blockchain based technology, a brief introduction about blockchain technology, problem description, challenges with blockchain for IoT and goals.

The rest of the thesis is organized in the form of different chapters where each chapter is concerned about different phases of this thesis. In this thesis, chapter 2 describes the related works which have been done for blockchain technology for IoT systems. Further, it followed by system model in chapter 3 that covers the architecture view of proposed framework for secured decentralized IoT based on blockchain based technology. In addition, chapter 4 describes the proposed framework focused on adaptive miner selection.

Moreover, chapter 5 describes the performance evaluation of adaptive miner selection in comparison with existing method. Chapter 6 concludes the paper with future work and recommendations.

2 Related Works

In [14] Ali et al proposed, a case study to use blockchain for a smart home application. He proposed a lightweight blockchain that does not contain the proof of work execution which should be performed by the minor node for each transaction. As security is the one of the main pillars of blockchain technology and proof of work prevents attackers from the tampering with chain of blocks. Therefore, removing the proof of work from the blockchain reduces its strength.

In reference [15], Neisse discussed about contract based blockchain applications. He explained about data accountability, scalability, provenance and performance of the applications. As per Neisse's report, sensitive data that is being exchanged more frequently need high level of security and scalability. However, a proper blockchain structure was not discussed in a detailed level in the report that can meet all requirements related to accountability, scalability and provenance. In reference [16] author presented, a private blockchain which is decentralized in nature. The main idea is to use blockchain with renewable energy and concept was to eliminate the proof of work from blockchain and use some voting procedure in order to verify the correctness of a block. Moreover, this approach was not good enough from security purpose.

In [17] Christidis investigated the blockchain technology and different consensus protocols. This investigation was focused on blockchain for IoT and provides a smart contract which has a unique set of rules. However, this smart contract is not an appropriate solution for tiny IoT devices. In [18] Crosby presented, an article about blockchain technology that describes about basic components of a blockchain and different financial and nonfinancial applications like notary, music applications and also about decentralized storage. In reference [19] author represented, a conceptual framework for smart home project that is using blockchain for sharing economic perspective. This framework generally explained the relationship between human, organization and technology itself.

In [20] Ouaddah explained, resource access policies in blockchain. It includes different types of transactions such as getAccess, delegetAccess and grantAccess. In this article Ouaddah advanced the blockchain architecture for smart homes/cities-based applications. He ensures the security and privacy of blockchain among IoT devices. In reference [21] author advocated a random miner selection-based consensus protocol.

In [22], a miner selection method is proposed that explains the benefits for a miner node throughout the entire blockchain network. This proposal also explains about corresponding elimination of computational overhead as a power in case of miner node. However, as per this proposal there is a chance that an inefficient node can be selected as a miner node with random miner selection procedure. Hence, to eliminate this problem this thesis focused on a machine learning based miner selection algorithm that is selecting miner node based on their performance.

In [23], a blockchain based secure framework was presented to store information from smart city based IoT systems. This framework contains multiple layers for example physical layer that contains IoT devices, next layer in the framework is communication layer. This layer includes information about communication protocols such as Bluetooth and 6LoWPAN. The last layer of this framework is distributed database layer that consists of blockchain implementation for IoT and user interface. This framework did not discuss about the basics building blocks of blockchain. It also did not provide any direction for the management of large amount of data from IoT devices in the blockchain. In [24] Eval et al. proposed, a different blockchain consensus protocol. As per the proposal the new protocol is scalable enough and called bitcoin next generation (Bitcoin NG).

Bitcoin NG described in [24], a leader node is selected with the help of a key block by using proof of work technique. This leader node is more like miner node that is used to process the proof of work for each transaction into the blocks and these blocks is called as micro blocks. The proposed method reduces the network propagation latency of transactions but process of selecting the leader node among all nodes in the network is more energy consuming.

3 System Model

In Figure 1, it shows system model/framework for this thesis that explains different IoT applications connected with the blockchain network to store the IoT applications data.

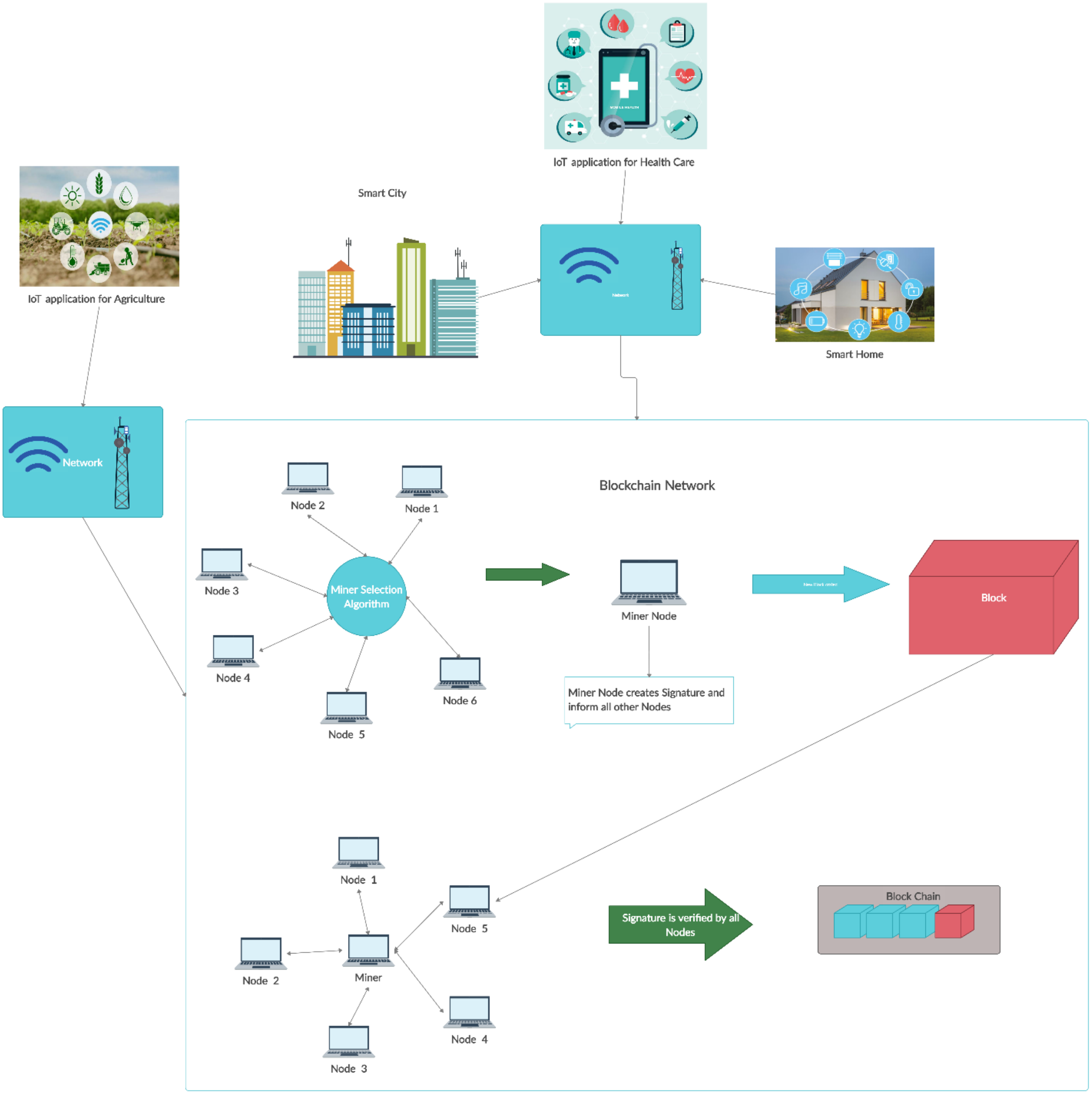

IoT application for Health Care
Smart City
IoT application for Agriculture
Network
Smart Home
Network
Blockchain Network
Node 2
Node 1
Node 3
Miner Selection Algorithm
Node 4
Node 5
Node 6
Miner Node
New Block created
Block
Miner Node creates Signature and inform all other Nodes
Node 1
Node 5
Node 2
Miner
Node 3
Node 4
Signature is verified by all Nodes
Block Chain

Figure 1.Blockchain used for decentralized IoT.

The main goal of this thesis is to provide a framework for secured decentralized IoT based on blockchain based technology. As we can see in Figure 1 several different IoT applications for example IoT applications for agriculture, health care based IoT applications and IoT applications for smart home and smart city. These applications consist of number of IoT devices such as sensors, actuators, micro controller. All these IoT devices are connected through a communication network that can be Wi-Fi, Bluetooth or any other network. IoT devices are using blockchain to store their data which are coming through the network connected to the IoT systems.

As blockchain stores data in the form of blocks. For each new block of information, the miner node is selected based on machine learning based miner selection algorithm. Miner node is used to generate the hash and create a signature to add new block to the blockchain. Miner node broadcasts this signature information to all other nodes in the network and once signature is verified by all other nodes in the network the new block is added to the blockchain. In the end, miner node is updated by itself and the contract is sent to the receiving node. The miner node is potentially most powerful node in terms of processing and memory as compared to other nodes. This thesis used consortium blockchain during this model for secured IoT systems [1].

Best miner selection plays an important role with blockchain technology for IoT devices. Network latency can be considered as a parameter to measure the performance of blockchain with IoT systems. Network latency is the collection of propagation latency, processing latency and queuing latency. Let's discuss about propagation latency first, propagation latency is proportional of the distance between network gateway and a miner. It is the time that data takes to propagate from network gateway to miner node [25]. For instance, to transfer x block from gateway network to y node, propagation latency will be denoted as:

$$PL(i,j) = \frac{D_{xy}}{s}$$

where,

'$D_{x\,y}$' is distance between the network (works as a gateway between IoT system and blockchain network) and miner node.

'S' is the propagation speed of communication channel.

Communication latency is the time that is taken by the network gateway to bring all bits of data for a single block on the channel that network is using to transfer data to the miner node. If network is bringing data for xth block to the yth miner node then communication latency can be defined as:

$$CL(x, y) = \frac{\gamma\mathrm{i}}{B_y}$$

Where,

γi = Amount of the data in xth block.

'B_y' is the bandwidth of communication between the network gateway and miner node.

Processing latency is related to the miner node. PrL (processing latency) of a block is the time that a miner node takes to generate target hash for that block. Hence, PrL is the time used to generate target hash by the miner node. PrL for generating target hash for xth block by yth miner node can be represented as follows:

$$PrL(x, y) = \frac{\mathrm{d} * 2^{32}}{HR_y}$$

Where,

d = current difficulty level.

HR_y = number of cryptographic hash operations performed by the miner node (y) in a second.

Queue latency can be defined as time that a block wasted in a queue waiting to be proceed by the miner. Queue latency for block x can be defined as follows:

$$QL(y) = \sum\nolimits_{n=1}^{M_b^y} PrL\ (n, y)$$

Where,

M_b^y = is the total number of blocks waiting to be executed in y^{th} miner

N = number of blocks

$PrL\ (n, y)$ = calculate processing time for n block

Hence, network latency of generating hash for x^{th} block using j^{th} miner can be calculated as follows:

$$N\,L\ (x, y) = PL\ (x, y) + CL\ (x, y) + PrL\ (x, y) + QL\ (y)$$

Machine learning based miner selection algorithm can improve the blockchain performance with IoT applications for queue latency, that will be a good impact on overall network latency. This thesis proposes a framework that can help to make blockchain more capable to store data for IoT applications [26].

4 Proposed Method

This thesis proposed a framework for secured IoT based on blockchain based technology which main consideration is machine learning based miner selection process. As I discussed in system model chapter that best miner selection can reduce the overall network latency for blockchain with IoT systems. In detail, this section explains blockchain, types of blockchain, machine learning for miner selection, LTR algorithm, and about implementation of proposed method with training and testing data.

Blockchain:

Generally, blockchain is a peer to peer database which is continuously growing. Data stored in blockchain in the form blocks, that contains timestamp, hash value of previous block and set of transactions data. All nodes in a blockchain network keeps same copy of data or blocks and each block is verified by each node before adding it to the blockchain. This blockchain database is decentralized in nature and called as ledger [25]. Each computing node in the blockchain network can be considered as a computer that runs a client software which connects it with the blockchain network. Blockchain is a distributed, decentralized and transparent ledger that provides secure and efficient transactions between the nodes. Moreover, blockchain has a set of specific rules those are used to add more data into the blockchain and transfer those data between nodes, i.e., smart contract, and a number of consensus algorithms for instance proof of work [27]. First block in the blockchain is called as genesis block. It does not contain any parent block.

Block:

A block can be considered as a container in the blockchain that is used to store the data. Blockchain contains several blocks and first block called as genesis block. Figure 2 illustrates the architecture of a block in the blockchain.

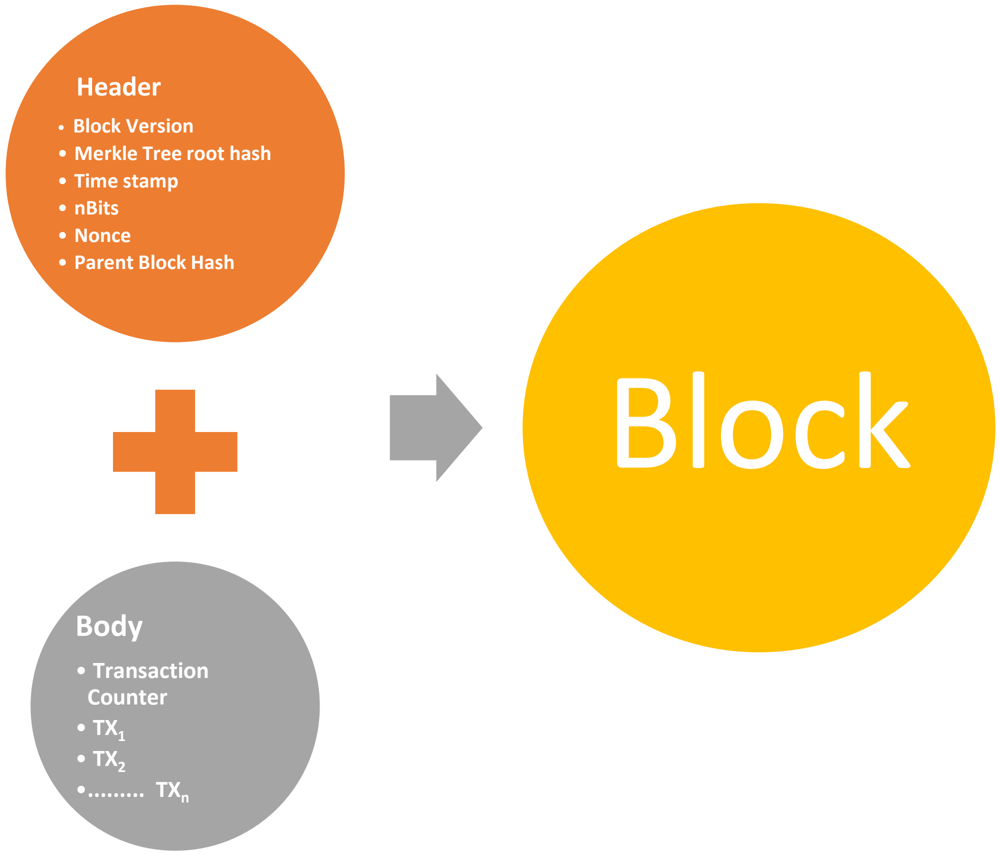


Figure 2.Block's architecture.

Moreover, a block is divided into two parts as you can also see in the given Figure 2. First part of the block is called as block header and second part is the body of block. Block's header contains block version, Merkle tree root hash, timestamp, nBits, nonce and parent block hash [25]. Moreover, block's body contains counter for each transaction and transactions itself. In more details, each section of complete block is described as follows:

Block version: Block version indicates the set of rules for block validation. Each block has a set of rules used for block validation.

Merkle tree root hash: Merkle tree also called as hash tree. It is a generalization of hash list and hash chain. Every leaf node in a Merkle tree stores hash value of a record or block from blockchain [28]. Merkle tree provides a high level of security in verification of data with large data structure. Here, in a block's header Merkle tree root hash section contains the hash value of all transactions in that block.

Figure 3 shows the Merkle tree architecture where leaf node has hash code of transactions 1,2,3 and 4.

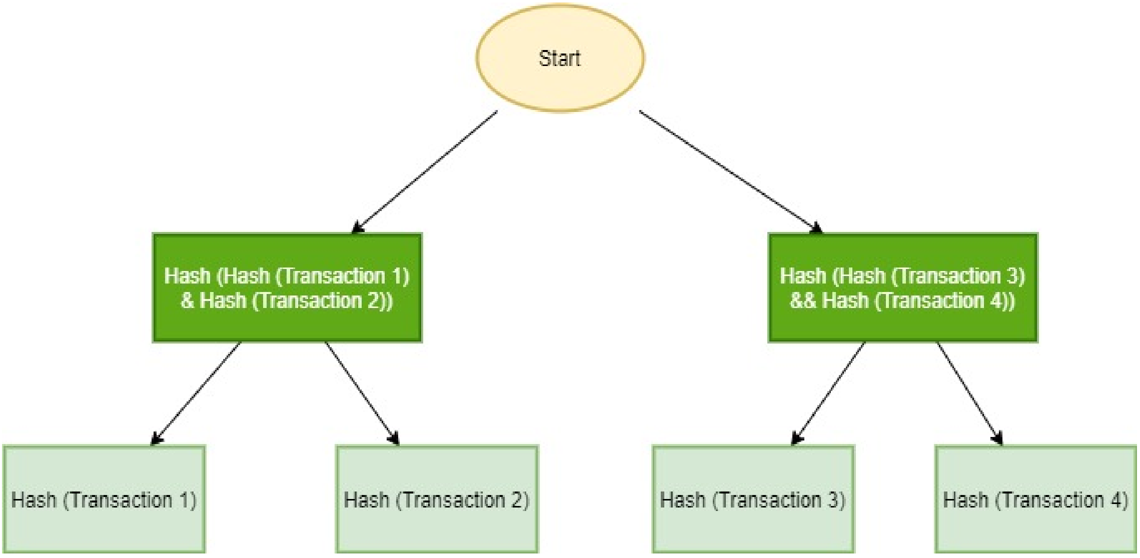

Start
Hash (Hash (Transaction 1) & Hash (Transaction 2))
Hash (Hash (Transaction 3) && Hash (Transaction 4))
Hash (Transaction 1)
Hash (Transaction 2)
Hash (Transaction 3)
Hash (Transaction 4)

Figure 3.Merkle tree architecture.

The parent node of leaf node stores hash code of leaf node. As we can see in Figure 3, left parent node stores hash code of leaf nodes those contain hash code of transaction 1 and 2 and right parent node stores hash code of leaf nodes those contain hash code of transactions 3 and 4. Hence, it is managing the hashes of transactions in this order.

NBits: is target threshold used for block's hash validation.

Nonce: is a value that is used to validate the hash for each block. Generally, nonce size is 4 bytes, it starts with 0 and increases with each hash calculation.

Parent block hash: Header of each block in the blockchain contains a 256-bit hash value that point to the previous block [6].

Moreover, body of a block contains number of transactions which depends on the size of block and size of each transaction. Digital signatures used by blockchain that uses asymmetric cryptographic algorithm.

Digital Signature uses asymmetric cryptographic algorithm to authenticate the digital documents or messages. It provides an extra layer to secure the messages or documents on communication channel. It commonly used in document verifications, for software distribution, financial transactions, contract management and in many other cases where integrity is on highest priority. Digital signature provides layered security for messages or documents sent through a communication medium. Basically, digital signature is equal to the traditional handwritten signature, but it is far more difficult to cheat with digital signature as compare to the handwritten signature. Moreover, digital signature has a cryptographic value generated from the signed data and a secret key that is known only by the signer of document. Digital signatures are based on public key cryptography [29].

During the digital signature process each user has two keys where one is the public key and second is private key. Private key is a confidential key that is used to sign the transactions. Once private key signed the transaction then signed transaction is broadcasted throughout the blockchain network. Typical digital signature process is divided into two phases where one is signing phase, and another is verification phase. During the signing phase, the sender signed the data with his/her private key and sent it to the receiver while during the second phase, receiver validate the value of data with the sender's public key in order to detect the data tempering. Elliptic curve digital signature (ECDSA) is the typical digital signature algorithm that is used by the blockchain technology [30].

Key characteristics of Blockchain

In brief, blockchain has following characteristics:

- **Persistency:** In the blockchain, it is impossible to delete or temper the transaction. Each transaction is validated by every node, block with invalid transaction could be discovered immediately and rejected by the nodes.

- **Decentralization:** Blockchain uses consensus algorithm to maintain the data consistency in the distributed network while in traditional transactions, third party applications were needed to validate the transaction. Moreover, traditional centralized transaction had risk of server down, and different kind of cyber attacks for instance denial of service (DoS) that is overcome by decentralization in blockchain technology.

- **Auditability:** Bitcoin blockchain used UTXO (Unspent Transaction Output) model to store the data about user's balance. This model helps in transaction tracking and verification because as per UTXO each transaction has to refer some previous transactions.

In the beginning this technology was developed for digital currency system to solve the double spending problem without any third-party security service. However, public blockchain performed extraordinary in case transactions in an unsecure environment but it also shows a few limitations in case of IoT. Indeed, blockchain has many points for instance transaction volume scalability and system responsiveness those need to be improved and considered as a shortcoming of blockchain for most of corporate applications. In order to overcome these shortcomings two different types of blockchain developed and currently it is categorized into three types: public blockchain, private blockchain and consortium blockchain [6].

Taxonomy of Blockchain systems:

Public blockchain is completely public and anyone can take part in the consensus process of this blockchain. All records are openly visible to everyone. In public blockchain, it is nearly impossible to tamper with the transactions because data is saved on many participates nodes [31]. Public blockchain is completely decentralized and open to the world. Moreover, in terms efficiency it does not have a good throughput due to large number of nodes. As result public blockchain has a high-level of latency.

Differently, **consortium blockchain** could be more efficient than public blockchain as it has comparatively a smaller number of validator nodes. Consortium blockchain uses cluster of pre-selected nodes that participates for consensus process. It can be considered as partially decentralized because it is constructed by number of organizations and only small portion of nodes selected for consensus process. In terms of security, transactions can be tempered easily with consortium blockchain because only few nodes are participating the validation process. Consortium blockchain could be used for different business applications. Recently, Ethereum also provided tool to develop the consortium blockchain and hyper ledger is a developing framework for consortium blockchain [32]. **Private blockchain** is fully centralized blockchain and nodes from a specific organization allowed to participate in the consensus process. It could be tempered easily as it is fully controlled by any single organization and all permission related to the nodes are dependent on the organization's decision [33]. Hence, private blockchain can give better performance in case of efficiency but it has risk with the immutability.

Table 1 described comparison between public, private and consortium blockchain with respect to different properties for instance read permissions, transaction throughput, security, network scalability, architecture, consensus procedure, consensus determination, and example of each technology. In comparison, public blockchain is more secure than the private and consortium blockchain because there are huge number of nodes participating in consensus process for public blockchain and it is quite impossible to temper data within a huge network.

Table 1.Attributes of public, private and consortium blockchain.

Attributes	Public Blockchain	Private Blockchain	Consortium Blockchain
Read Permissions	Publicly available, anyone can access.	Could be restricted or public within the organization	Could be restricted or public
Transactions Throughput	Low	High due to less numbers of nodes	High due to less numbers of nodes
Security	High	Low	Low
Network Scalability	High	Low to medium	Low to medium
Architecture	Highly- decentralized	Partially decentralized	Centralized
Consensus Procedure	Any node can participate so process is permission-less.	Only few pre-selected nodes can participate. Hence process is permissioned	Permissioned
Consensus Determination	All miners	Only few selected nodes	Nodes from a single organization
Example	Bitcoin, Ripple etc	HyperLedger, Ethermint, Tendermint, Quorum etc.	

Moreover, the comparisons show that private blockchain is an alternative of public blockchain for small organization level applications. Public blockchain is properly decentralized while consortium blockchain is partially decentralized and private blockchain has a centralized architecture.

4.1 Machine Learning based Miner Selection

Machine learning is based on the concept of models which can study from data, pick out patterns, and can make selections with minimal human interventions. Machine learning is basically based on algorithms that learns from practice and improve through the experience. These algorithms are used to create a mathematical model which is based on sample data that is also called "learning data" and used to make predictions. Intensively, machine learning is related with the computational statistics, that focuses on computer forecasting. Machine learning is growing day by day with modern technologies, earlier it was more focused on pattern recognition and theory, to make computer learn without performing the actual task [34]. Currently, machine learning is more focused iterative function and learning from data. Machine learning algorithms used in lot of different applications. Machine learning itself is divided into different types such as supervised learning, semi-supervised learning, unsupervised learning and reinforcement learning etc.

In **supervised learning**, both example input, and desired output are predefined, and model need to be trained for a predefined input and output. The training data used for supervised learning contains both input and output. Sometimes input dataset is a vector that refers to a single output value for input vector. Hence, supervised learning method analyse the training or sample data and prepare a method, which can be used later for predicting output for new input data [35].

Unsupervised learning is another type of machine learning approach that is used to train a data model without any pre-existing labels. In unsupervised learning, training data only have input data there is no desired output data available in the sample dataset. Unsupervised machine learning has minimum human supervision, algorithm finds all undetected existing patterns from the dataset and prepare a model that is used to predict out or make decision for output for testing data. Supervised learning generally used principal components and cluster analysis to find all existing patterns in the give dataset. Clustering is used for grouping or making dataset for better recognition of relationships in the training dataset [36].

Semi-supervised learning comes into approach somewhere between the supervised learning and unsupervised learning. In semi-supervised learning, training dataset contains both labelled and non labelled data where the amount of non labelled data is comparatively high in the dataset. Semi-supervised learning algorithm generally used in the situations where labelling process is more costly. In such cases semi-supervised performed a great work with small amount of labelled and huge number on unlabelled training data [37].

Reinforcement learning does not need any labelled data, so it is completely different than supervised machine learning. Reinforcement learning is more focused on finding a balance between exploration and exploitation and stated in the form of Markov decision process (MDP) [38]. The main difference between traditional dynamic programming methods and reinforcement learning is that it predicts exact mathematical model of MDP instead of targeting largest MDP's

In a decentralized network, miner selection is a process of selecting best nodes from the set of nodes. During complete miner selection process, all nodes compete to be miner to generate the target hash for each new block in order to prevent data tampering in the blockchain. Proposed framework focused on a method to select the best miner based on their performance. Generally, miner selection process is a random process that is also shown in the given Figure 4. It can be denoted as Random Miner selection (RMS). As we can see from the given figure below, there are n number of nodes approaching the network manager to be a miner node. Each node has its own properties, but those properties are not separately evaluated by the network manager or any other procedure. Network manager is selecting a miner randomly based on the timing of approaching by nodes in order to be a miner node.

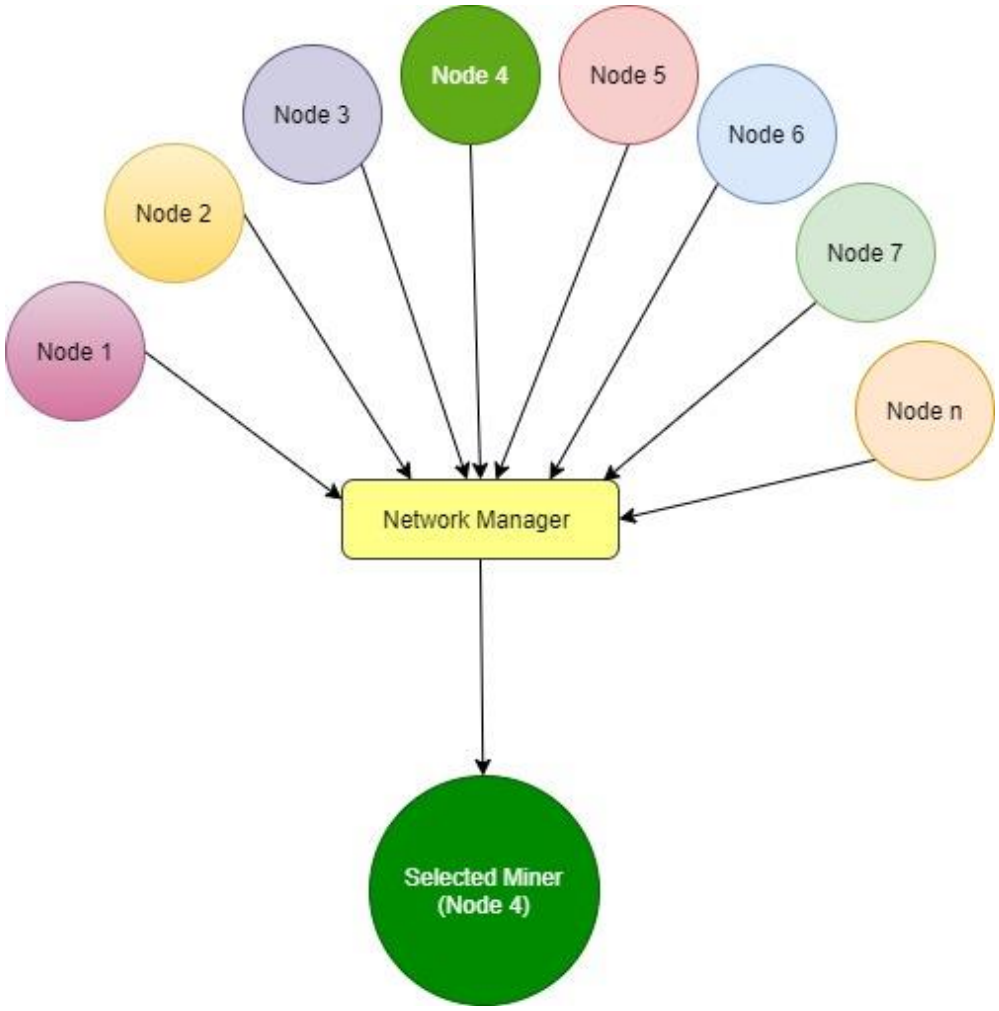


Figure 4.Random miner selection.

Any random miner is selected by the network manager that is used to run the consensus for block. Miner node need to be strong enough to do proof of work for new block as it consumes huge processing power. A competent miner plays a vital role in the blockchain and miner selection process is quite time consuming when there are many nodes in the blockchain.

In order to simplify, miner selection process this thesis proposed to rank each node in complete blockchain network based on their properties using LTR. The thesis considers three properties to rank each node for instance computing power, hash generation rate and block propagation delay. By using this approach of ranking nodes, thesis will rank each node with a number based on there properties and the node with most desirable properties will be on the top in the ranking list and selected as a miner node.

LTR is a methodology that applies on supervised machine learning in order to resolve the ranking problems. Supervised machine learning already explained above in this thesis. As it solves prediction problems with classification or regression on a signal instance at a time. The main goal of traditional machine learning algorithm is to come up with a single decision for example yes or no or a numerical score for an instance. Here, LTR is applied on top of traditional supervised learning algorithm to solve ranking problem on a list of items [39]. LTR provides a proper ordering of all desired items. Search engine ranking is the most common example of LTR, but it can be used anywhere when we need a ranked list of items instead of single value at an instance.

Supervised Learning
Ranking
Learnign to Rank (LTR)

Figure 5.Learning to Rank with supervised machine learning.

Figure 5 describes, how LTR is applied on top of traditional machine learning algorithm. The main benefit with LTR is that it solves problem over a list of items while supervised machine learning only worked over a single instance at a time. LTR is more focused on relative ordering among all the items instead of getting the exact score for each item. The general architecture of LTR is described in Figure 6.

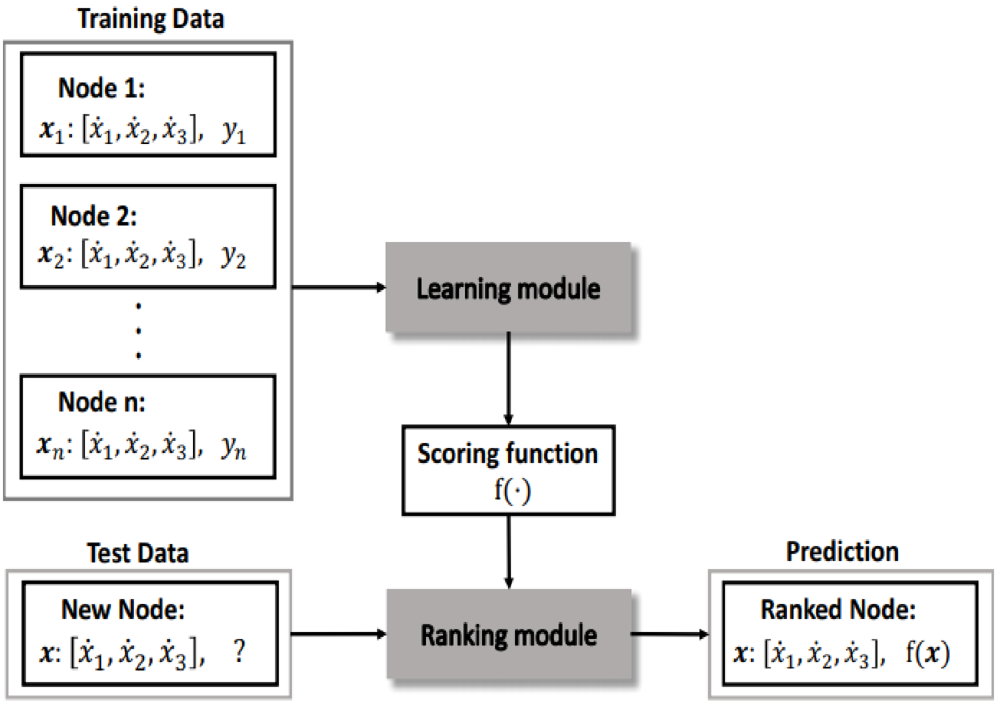

Training Data
Node 1:
x_1: [$\dot{x}_1, \dot{x}_2, \dot{x}_3$], y_1
Node 2:
x_2: [$\dot{x}_1, \dot{x}_2, \dot{x}_3$], y_2
Node n:
x_n: [$\dot{x}_1, \dot{x}_2, \dot{x}_3$], y_n
Learning module
Scoring function
f(·)
Test Data
New Node:
x: [$\dot{x}_1, \dot{x}_2, \dot{x}_3$], ?
Ranking module
Prediction
Ranked Node:
x: [$\dot{x}_1, \dot{x}_2, \dot{x}_3$], f($x$)

Figure 6.Learning to Rank architecture. [40]

LTR algorithm works on top of supervised learning so it works on labelled training dataset. Training dataset used by the learning module to learn by finding the relationship between input and output. Moreover, data passed through a scoring function that provides a score for ranking the list of items and this scoring function is used by ranking modules for ordering the predicted data in a desired way. For instance, in case of searching engine ranking model used a set of queries those are generated based on the ratings provided by the users [40]. In case of blockchain, LTR resolves the issue with the miner selection that makes blockchain more advanced to be used with IoT systems.

4.1.1 Applied LTR for Adaptive Miner Selection

In order to simplify the miner selection process in the blockchain, this thesis applies learning to rank algorithm to select the best miner based on three important properties. Here these three properties described in detail, **computing power** for an ideal miner should be high because miner node needs to do consensus and generate hash for each block that requires huge processing power. **Hash generation rate** depends on the computing power of the system or node and it must be good specially in case of blockchain with IoT when transaction is more frequent. **Block propagation delay** is the time taken by a new block to reach through each node in the network. Each node verifies the new block and add it to its own blockchain. Verification performed by each node is time taking process. Hence block propagation delay should be the minimum for an ideal miner node.

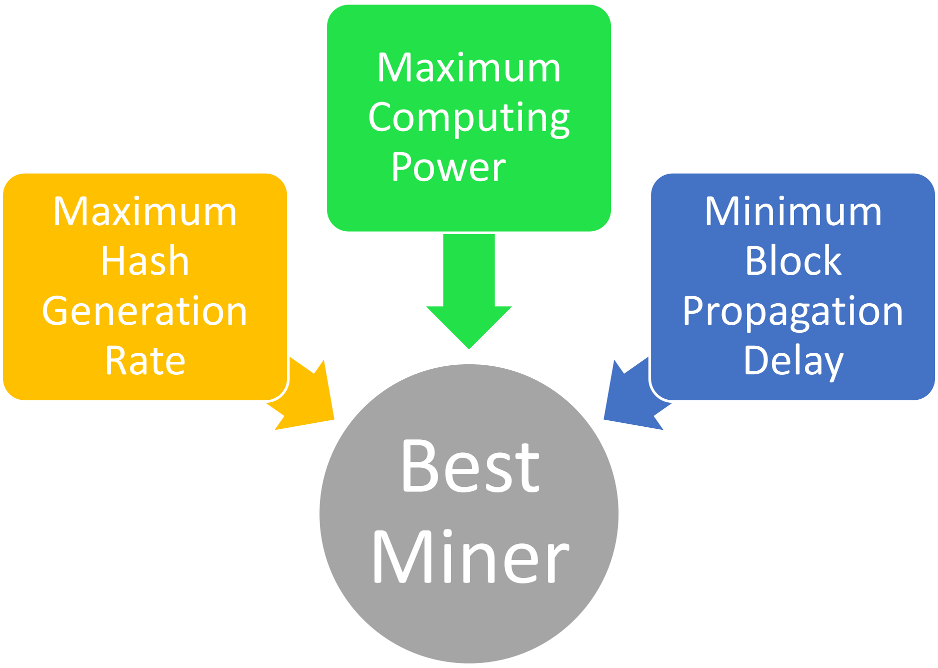


Figure 7.Best miner' s properties.

Figure 7 shows, properties of an ideal miner node. These properties have been used by the LTR algorithm in order to predict the best miner. LTR is taking training data with number of miner nodes with different combination of properties and ranking these nodes in desired order with maximum computing power and hash generation rate and minimum block propagation delay.

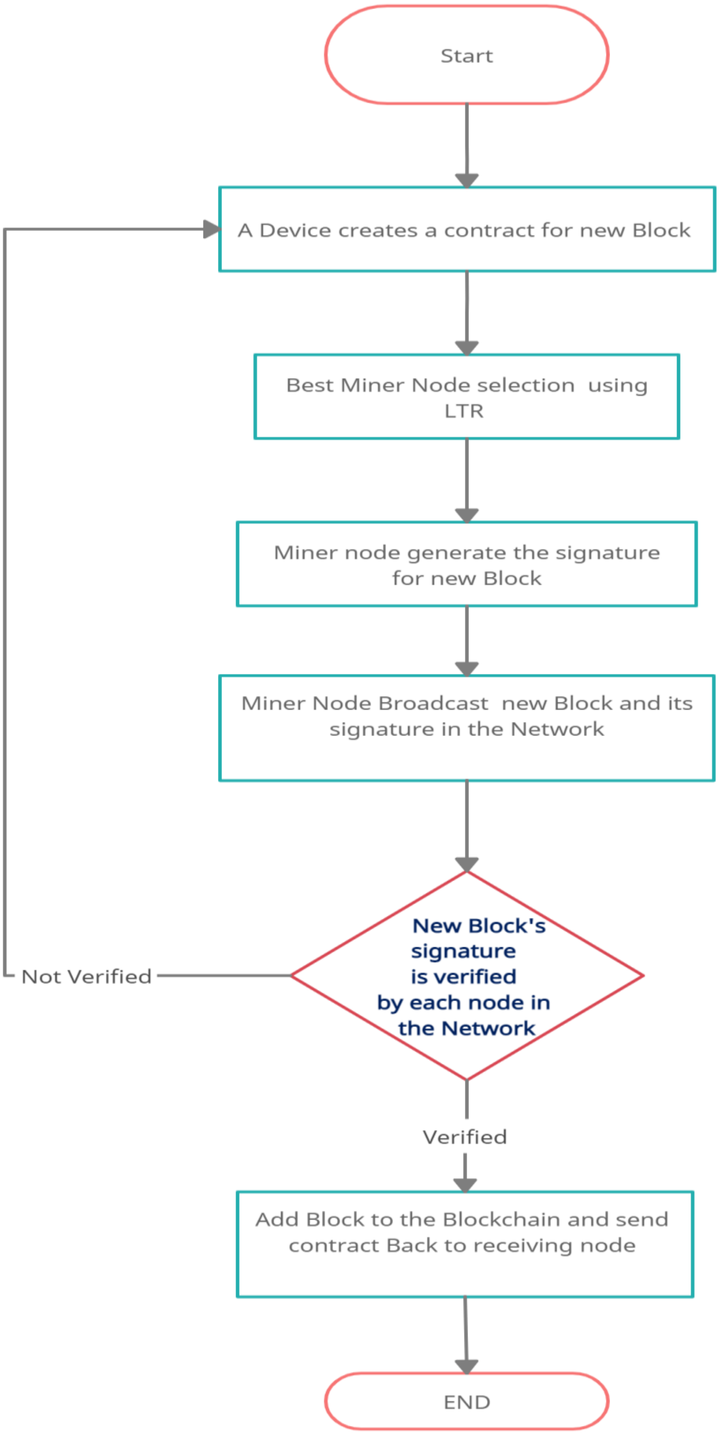

Start
A Device creates a contract for new Block
Best Miner Node selection using LTR
Miner node generate the signature for new Block
Miner Node Broadcast new Block and its signature in the Network
New Block's signature is verified by each node in the Network
Not Verified
Verified
Add Block to the Blockchain and send contract Back to receiving node
END

Figure 8.Blockchain’s flow chart with adaptive miner selection.

Figure 8 shows, the flowchart for blockchain operations with proposed miner selection method with LTR algorithm. Miner node’s processing power is the main concern for hashing and consensus of the blockchain in the IoT. Hence, this thesis providing an improvement in block verification process as well with LTR based best miner selection method and in decentralized IoT, miner node with best processing capability improves the overall blockchain performance [6].

In the proposed framework, thesis using LTR algorithm in order select the best miner node. The training dataset contains data of 60 nodes where each node has three different attributes i.e. computing power, hash generation rate and block propagation delay. In LTR, first model is learning based on this training data and after learning module data is ranked based on the ranking function.

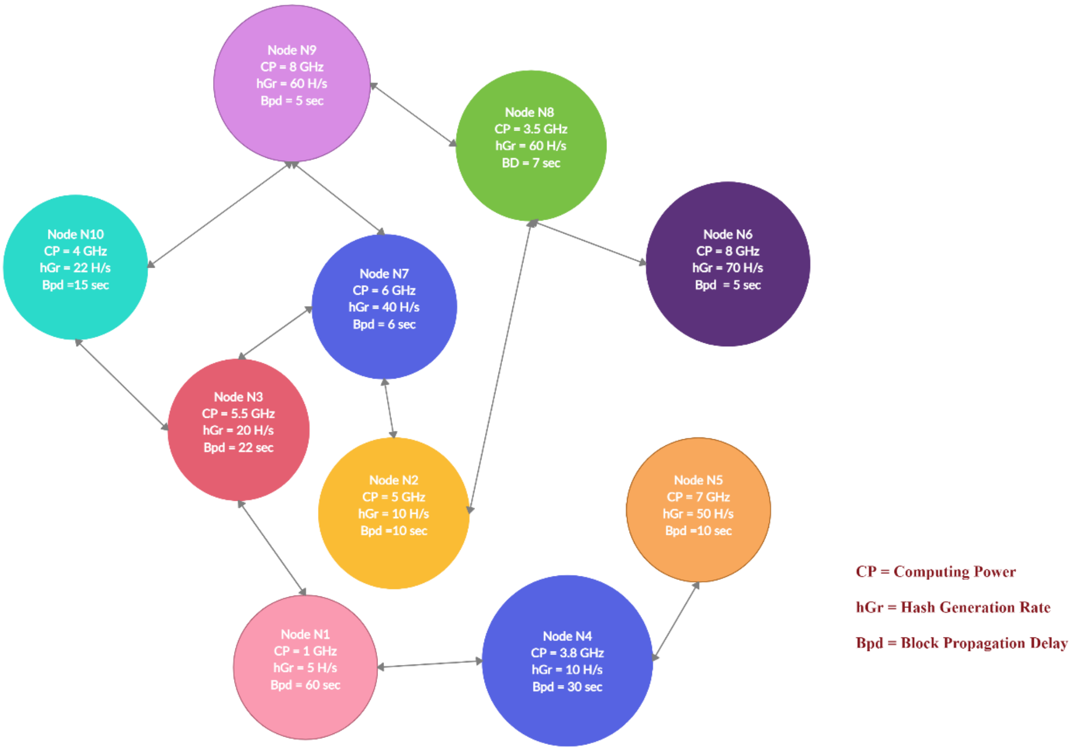


Figure 9.Nodes in the training dataset.

Figure 9 illustrates, an example of nodes that thesis used to train the model to select the best miner from all nodes in the network. computing power, hash generation rate and block propagation delay are properties of each nodes [5]. Here, I have a set of data from around 60 nodes to train the algorithm. Dataset is denoted as τ and

$$\tau = \{(X_1, y_1), (X_2, y_2), \ldots\ldots\ldots (X_n, y_{n,})\}$$

contains n does in the decentralized IoT network, where X_i is the i^{th} node in the network and y_i is the corresponding scored value associated with X_i and used as a label to train the model. Each X_i in the data set τ is a set, represented by a list of properties and can be denoted as feature vector with dimension denoted as d.

$X_i = [x_1', x_2', x_3']$ where x_i' is the i^{th} property of a node from the list of attributes considered to make decisions for a miner node. Three important attributes considered for node is computing power, hash generation rate and block propagation delay.

Here, y_i is the scored values calculated for each node based on the scoring function f(.) Scoring function f(.) for node X_j will be f (X_j). Scoring function used by the algorithm is:

$$f(X_i) = \frac{CP + hGr + (n - Bpd)}{n}$$

Thus, for any j^{th} network with n nodes the list of scores can obtained efficiently like:

$$S^j = [f(X^j{}_1) \ldots\ldots. f(X^j{}_n)]$$

After training the algorithm this scored value can be ranked in descending order, in order to have a best node with highest rank. In LTR, the scored value is sorted based on the ranking function *r(.)*. Ranking function is $r\,(S^j) = S_n > S_{n-1}$

After scoring all nodes with scoring function. Nodes are being sorted based on the ranking function and added to the miner pool. The top node from the miner pool selected as a miner node. Hence, this is the description about proposed method for adaptive miner selection with LTR.

5 Results and Discussion

This thesis focused on an adaptive miner selection methodology and evaluating it with LTR in the blockchain technology. For the simulation, the parameters are considered as hash generation rate, block propagation delay and computing power for each node. All three parameters are varying per node. Table 2 shows, parameters considered for the simulation. I used python programming language for simulation with scikit-learn framework for machine learning [41].

Table 2.Simulation parameters.

Parameter	Value
Number of nodes	60
hGr_{min}	1 H/s
hGr_{max}	100 H/s
Bpd_{min}	1 sec
Bpd_{max}	60 sec

This thesis considered, proof of work consensus for blockchain validation. As given in Table 2 total number of nodes considered is 60, minimum hash generation rate used for simulation is 1 hash per second where maximum hash generate rate for a node is 100 hash per second. Similarly, for block propagation delay, minimum value of delay is 1 second and maximum value is 60 seconds. Every node is scored based on scoring function ($f(X_i)$) which is described in chapter 4 and it can vary from 1 to 5 based on the properties of node.

In results, I compare the hash generation rate of the miner's node selected by LTR and RMS. In Figure 10, it shows the comparison of hash generation rate of different miners. Moreover, Figure 11 shows the comparison of block propagation delay of different miners. In addition, the learning performance of LTR is also described in Figure 12 which shows the graph plotted with training and testing data set.

The results of proposed method have been evaluated by three different ways described below:

- Evaluation based on hash generation rate
- Evaluation based on block propagation delay
- Learning to rank algorithm performance evaluation

5.1 Comparison with respect to Hash Generation Rate

In Figure 10, it shows the hash generation rate of different miner nodes after applying LTR and RMS based miner selection procedure.

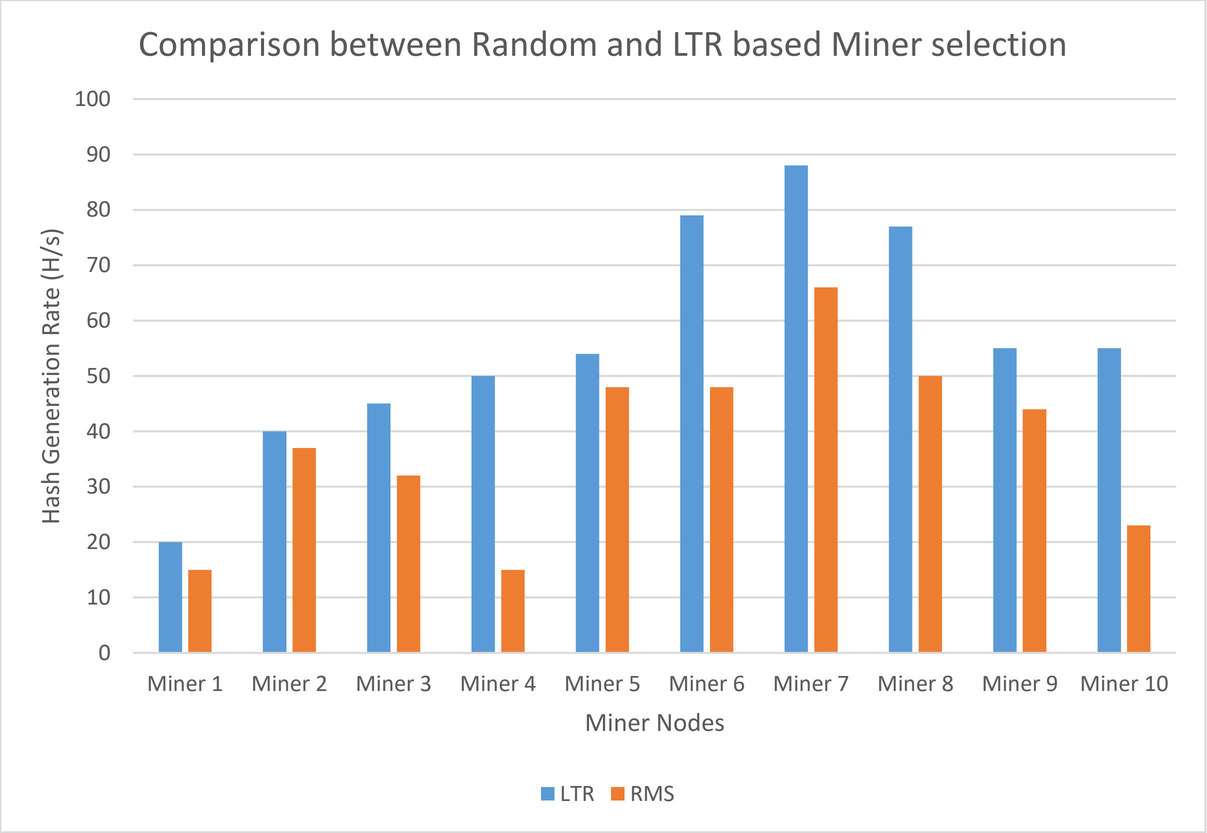


Figure 10.Hash generation rate with LTR and RMS.

As we can see the LTR based miner selection shows overall high hash generation rate than RMS procedure. I consider 10 nodes to examine the overall performance for hash generation of both procedures. We can observe that miner node with LTR has higher hash generation rate among all nodes while RMS is not performing well because RMS based selected miner is not evaluated based on its performance. In addition, the difference between hash generation rate is more than 30 H/s in case of 6th miner node. LTR trained to provide the miner node with high hash generation rate. It has a ranking for all available nodes in the network and top ranked node is only selected as a miner where RMS based approach is selecting any random node that is approaching to be a miner at earliest opportunity. Hence, from Figure 10, we can see that the overall performance of LTR algorithm is better than RMS based procedure in the blockchain.

5.2 Comparison with respect to Block Propagation Delay

Figure 11 shows comparison between RMS and miner selection with LTR with respect to block propagation delay for each miner node.

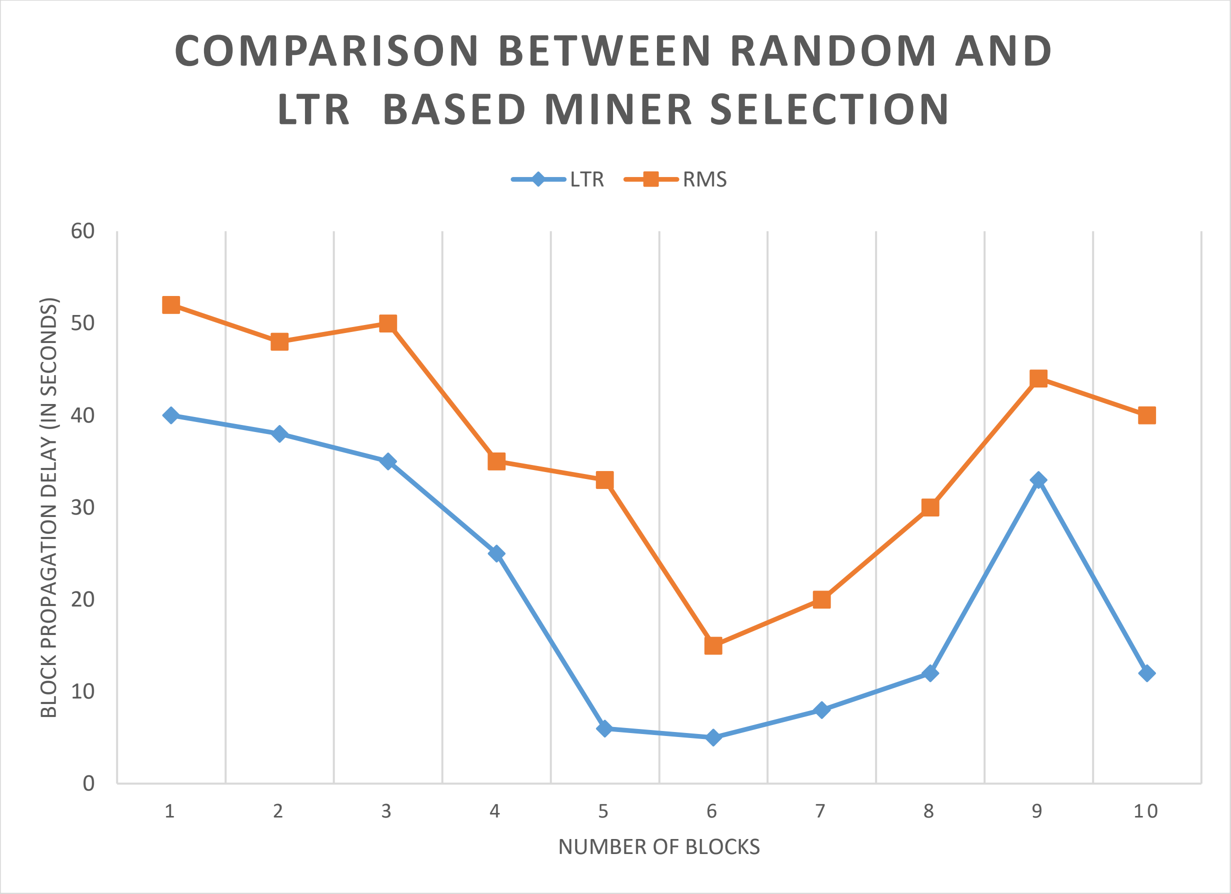


Figure 11.Block propagation delay with LTR and RMS.

Here, block propagation delay is the time taken by each new block to reach through in the network. Each node verifies the new block and then add it to its own blockchain. We can observe from the Figure 11 that block propagation delay is varying for each block for both LTR and RMS. In addition, LTR shows overall less block propagation delay as compare to RMS because miner selected based on LTR has better hash generation rate and computing power that decreases the block propagation delay. Hence, LTR algorithm perform exceptionally with respect to block propagation delay as compare to RMS.

5.3 Evaluation of LTR Performance

Figure 12 shows the overall LTR algorithm performance with training and testing dataset.

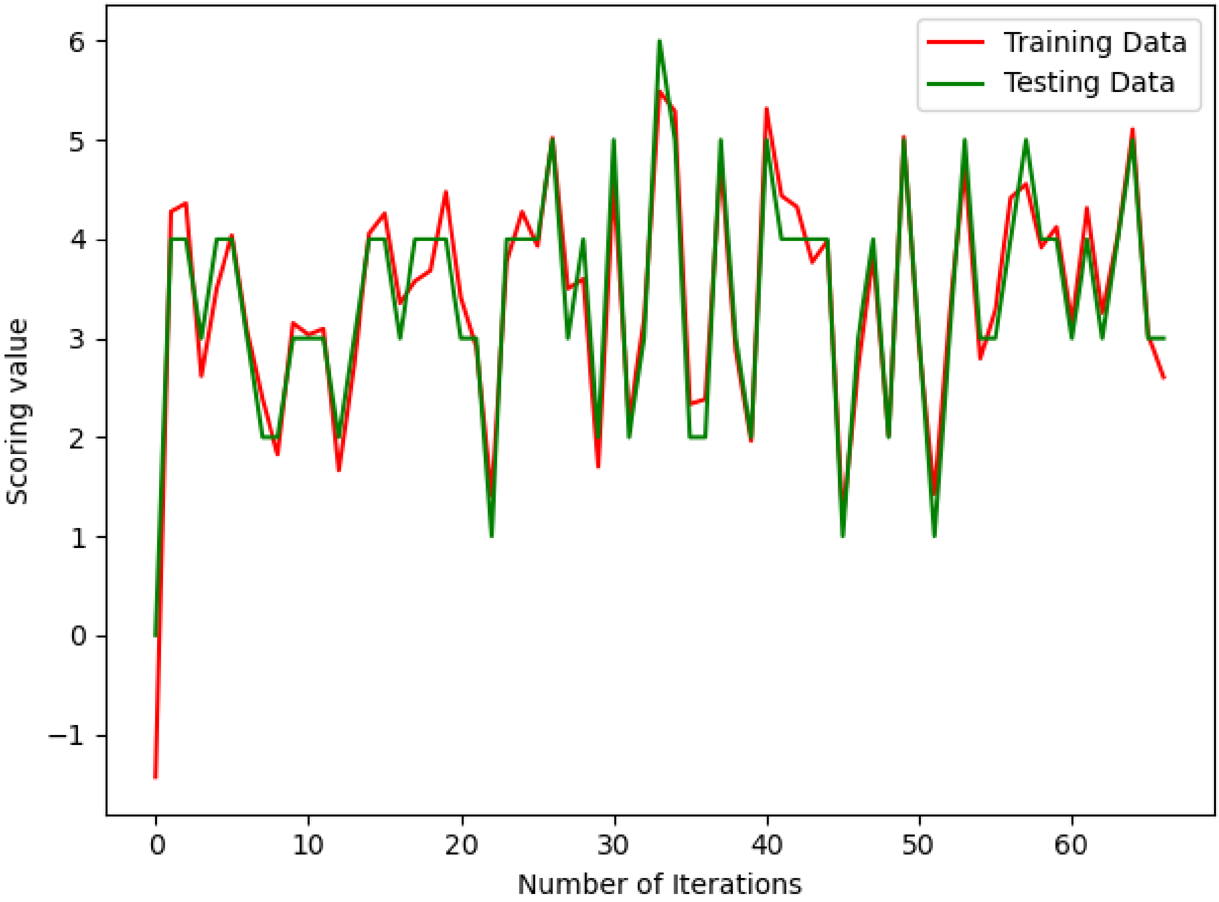


Figure 12.LTR performance over number of iterations.

As we can see from the graph training dataset has overall good performance with testing data set. I consider overall 60 iterations to examine the overall performance of this learning to rank algorithm. The LTR algorithm is learning based on the scoring function described in the proposed method section of this thesis. Moreover, graph shows variations in the scoring values over the number of iterations. These variations are based on the competency level of the node participating to be a miner and this competency level of each node is decided based on the computing power, hash generation rate and block propagation delay for each node. From Figure 12, we can see that the LTR algorithm has a good learning performance.

5.4 Discussion

From the above evaluation, we can see that the overall miner selection approach gets better with LTR algorithm. In Figure 10 and 11, I have compared my simulation results with respect to hash generation rate and block propagation delay for random and LTR based miner selection. Comparisons from Figure 10 and 11 shows that LTR gives outstanding performance over RMS.

In Figure 12, I provided the evaluation for LTR algorithm with training and testing data set. During the number of iterations, the scoring value is varying based on scored value of each node. Moreover, testing dataset shows that algorithm learnt efficiently from training dataset and responding with a better performance during the testing phase. As per the observations, LTR based miner selection improves the blockchain scalability or overall performance with IoT systems.

6 Conclusion and Future work

Currently, millions of devices have been connected with IoT systems and it is increasing on daily basis. Security is one the major concerned with IoT systems. Blockchain has a potential to solve the security issue with IoT systems. Blockchain is immutable and data cannot be tampered once it stored into a Blockchain. It is highly reliable, secure, completely decentralized, transparent, immutable and decentralized architecture to record the data as a public ledger. However, scalability is the major challenge with blockchain while we are considering blockchain in context of IoT because high computational power is needed to solve proof of work.

This thesis proposed a framework for secured IoT based on blockchain based technology that specially considered an adaptive miner selection method, LTR algorithm, which outperforms random selection in terms of computing power, hash generation rate and propagation delay.

Results of this work shows outstanding performance of blockchain with IoT systems. As shown in above result description LTR algorithm performance has been evaluated over 60 iterations and algorithms shows an outstanding performance. LTR provides ranking to each node with learning which helps to get best ranked miner for each new block. Adaptive miner selection improves block propagation delay for each block, hash generation rate for each new block. It makes the mining of new block more scalable with IoT system where transaction is more frequent.

In future work, we can implement hybrid consensus algorithm in this solution with adaptive miner selection algorithm which can improve overall blockchain performance with IoT at next level in terms of scalability, security and accessibility.

References

[1] P. A. K. M. V. J. Laplante, “Building Healthcare systems with IoT,” *IEEE Systems Journal,* p. 12, 2018.

[2] M. Rouse, “Internet of Things (IoT),” 2014.

[3] A. M. H. Montazerolghaem, “Load-balanced and QoS-Aware software defined Internet of Things,” *IEEE Internet of Things,* 2020.

[4] VECAP, “Modern IoT systems,” 29 January 2019. [Online]. Available: https://medium.com/vecap-next-generation-of-smart-home/modern-iot-systems-are-centralized-and-inefficient-34353761cb7b. [Accessed 19 November 2020].

[5] D. Z. Morris, “Leaderless, Blockchain- based venture capital fund raises,” 2016.

[6] M. L. K. R. Iansiti, “The truth about Blockchain,” in *Harvard University* , Cambridge, 2017.

[7] A. B. J. F. E. M. A. S. Narayanan, “Bitcoin and cryptocurrency technologies: a comprehensive introduction,” in *Princeton University Press*, Princeton, New Jersey, 2018.

[8] J. C. A. Brito, “Bitcoin: A primer for policy makers,” in *George Mason University,* Fairfax, 2013.

[9] S. Armstrong, “Move over Bitcoin, the Blockchain is only just getting started,” 2016.

[10] C. G. J. S. Catalini, “Some simple economics of Blockchain,” 2019.

[11] D. A. Tapscott, “Here’s Why Blockchains will change the world,” 2016.

[12] K. Bheemajah, “BlockChain 2.0: The Renaissance of Money,” 2016.

[13] A. J. F. Z. G. E. Sherman, “On the Origins and variations of Blockchain Technologies,” *IEEE Security Privacy* , pp. 72-77, 2019.

[14] S. K. R. J. a. P. G. A. Dorri, “Blockchain for IoT security and privacy: The case study of a smart home,” in *IEEE International Conference,* 2017.

[15] G. S. N.-F. R. Neisse, “A Blockchain based approach for data accountability and provenance tracking,” in *12th International Conference,* 2017.

[16] B. N. C. B. D. D. a. C. W. E. Mengelkamp, “A Blockchain-based smart grid,” in *Computer Science Research and Development,* 2018, pp. 207-214.

[17] M. D. K. Christidis, “Blockchains and smart contracts for the Internet of Things (IoT),” 2016, pp. 2292-2303.

[18] P. p. S. V. V. K. M. Crosby, “Blockchain Technology: Beyond bitcoin,” *Applied Innovation,* pp. 6-10, 2016.

[19] J. Y. K. Z. J. Sun, “Blockchain-based sharing services: What blockchain technology can contribute to smart citites,” *Financial Innovation,* p. 26, 2016.

[20] A. A. E. A. A. O. A. Ouaddah, Towards a novel privacy- preserving access control model based on blockchain technology in IoT, Europe and MENA cooperation advances in Information and Communication technologies, 2017.

[21] R. D. P. K. K. B. K. Peterson, “A Blockchain-based approach to health information exchange networks,” in *Proc. NIST workshop*, 2016.

[22] G. Greenspan, “Multichain private blockchain- white paper,” [Online]. Available: http://www. multichain. com/download/MultiChain-White-Paper. Pdf. [Accessed 12 October 2020].

[23] V. M. K. s. K. Biswas, “Securing smart cities using Blockchain technology,” in *IEEE 2nd international conference on Data science and Systems*, 2016.

[24] A. G. E. S. R. V. R. Eyal, “Bitcoin -ng: A scalable Blockchain Protocol,” pp. 45-59, 2016.

[25] S. Nakamoto, “Bitcoin: A peer to peer electronic cash system,” 2020. [Online]. Available: https://bitcoin.org/en/bitcoin-paper. [Accessed 11 September 2020].

[26] A. R. K. M. S. P. S. Midya, “Multi-objective optimization technique for resource allocation and task scheduling in vehicular cloud architecture: A hybrid adaptive nature inspired approach,” *Journal of network and computer applications,* pp. 58-84, 2018.

[27] S. K. Y. S. X. J. J. W. Y. Zhang, “Smart contract- Based access control for the Internet of things,” *IEEE Internet of things (IoT) Journal,* 2019.

[28] G. Becker, “Merkle Signature schemes, Merkle trees and their cryptanalysis,” in *Ruhr-University*, 2013.

[29] E. Paul, “What is Digital signature- How it works, Benefits, Objectives, Concept,” in *EMP Trust HR.*

[30] A. M. S. V. D. Johnson, “The elliptic curve digital signature algorithm (ECDSA),” *International Journal of Information security,* pp. pp 36-63, 2001.

[31] V. Buterin, “On public and private Blockchains,” 2019. [Online]. Available: https://blog.ethereum.org/2015/08/07/on-public-and-private-blockchains/. [Accessed 2020 October 20].

[32] “Hyperledger project,” 2015. [Online]. Available: https://www.hyperledger.org/. [Accessed 12 October 2020].

[33] C. c. developmen, “github,” [Online]. Available: http://github.com/ethereum/wiki/wiki/Consortium-Chain-Development. [Accessed 22 October 2020].

[34] J. R. F. H. A. D. K. M. A. Arthur Samuel, “Without being explicitly programmed,” pp. 151-170, 1996.

[35] P. N. Stuart J. Russell, “Artificial Intelligence: A modern approach,” in *Prentice Hall*, 2010.

[36] G. S. T. Hinton, “Unsupervised Learning: Foundations of Neural computation,” in *MIT Press*, 1999.

[37] O. B. Z. A. Chapelle, “Semi-supervised learning,” in *MIT Press*, Cambridge, 2006.

[38] L. P. L. M. L. M. A. W. Kaelbling, “Reinforcement Learning: A survey,” *Journal of Artificial Intelligence Research,* pp. 237-285, 2001.

[39] L. Tie-Yan, “Learning to Rank for information Retrieval,” 2019.

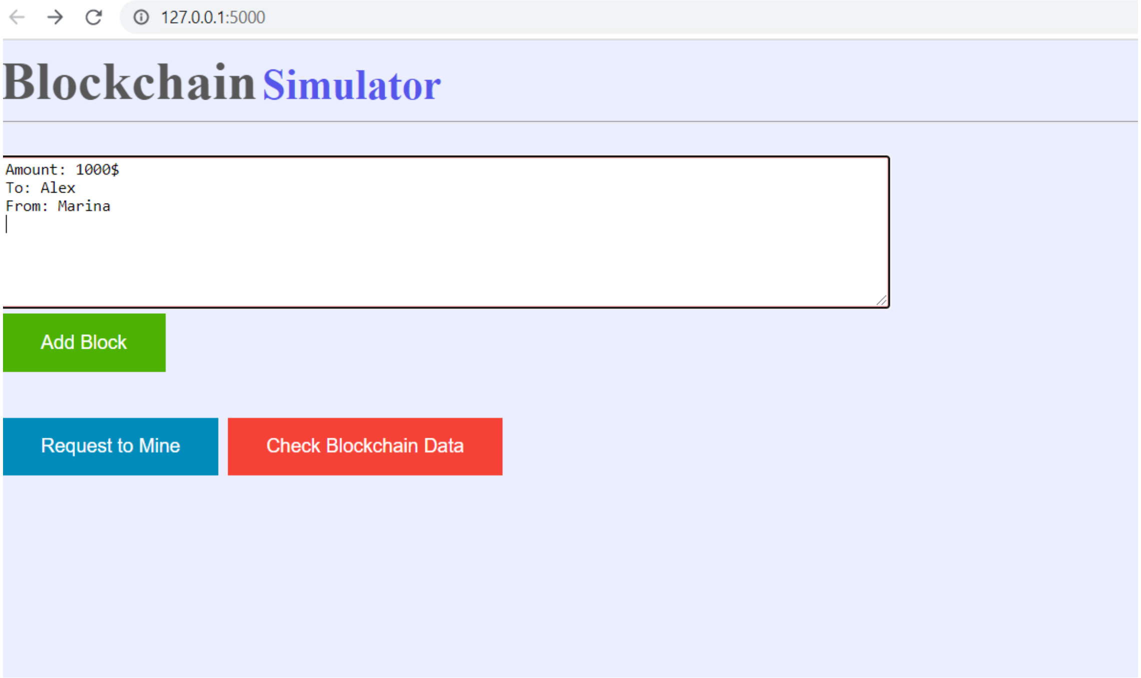
[40] B. Nikhil, “ Medium,” 2018. [Online]. Available: https://medium.com/@nikhilbd/intuitive-explanation-of-learning-to-rank-and-ranknet-lambdarank-and-lambdamart-fe1e17fac418, 2016. [Accessed June 2020].

[41] P. e. al, “Scikit-learn: Machine Learning in Python,” 2011. [Online]. Available: https://scikit-learn.org/stable/about.html#citing-scikit-learn. [Accessed 1 August 2020].

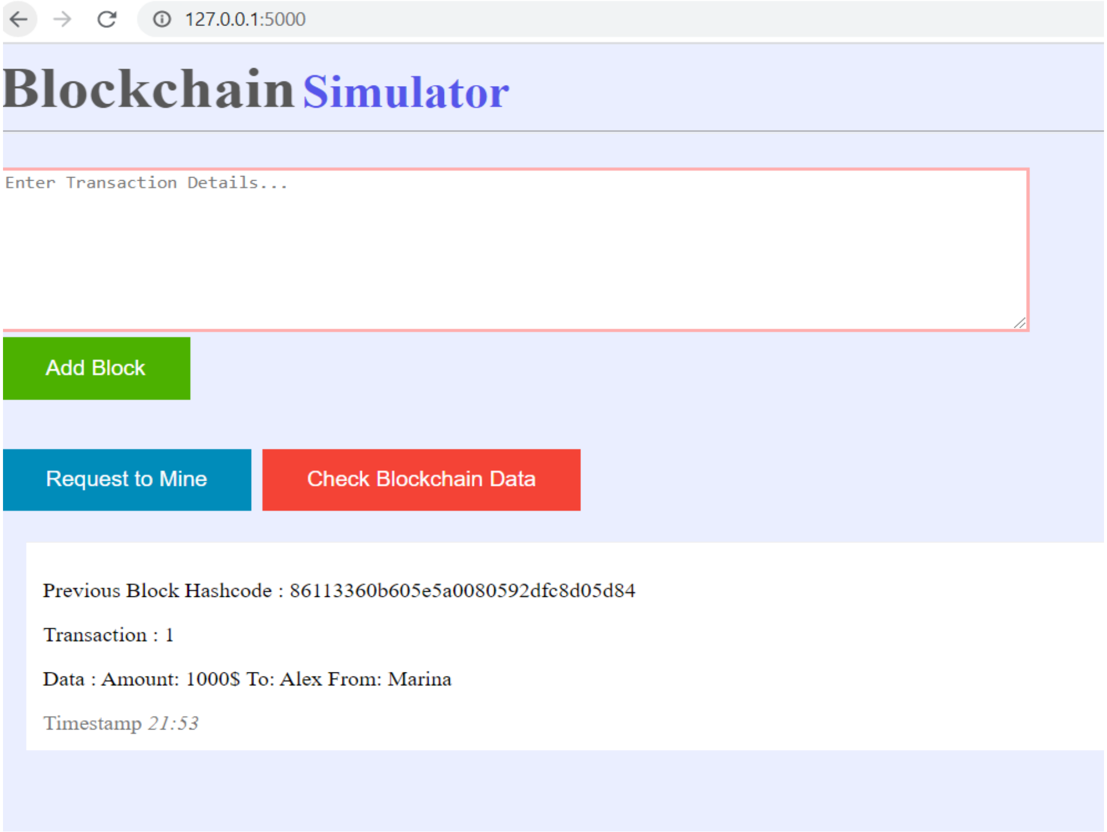
Appendix 1 Source

```
from hashlib import sha256
import json
import time
import pandas as pd
from sklearn import preprocessing
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from flask import Flask, request
import requests

class Block:
def __init__(self, n_r, tx_data, adding_time, preblock_hash, nonce_val=2):
        self.n_r = n_r
        self.tx_data = tx_data
        self.adding_time = adding_time
        self.preblock_hash = preblock_hash
        self.nonce_val = nonce_val

    def select_miner(self):
        existing_nodes = 'nodes_data.csv'
        nodes_pool = pd.read_csv(existing_nodes)
        x = nodes_pool.drop(['Result', 'RankData'], axis=1)
        y = nodes_pool['RankData']
        lab_enc = preprocessing.LabelEncoder()
        encoded_Y = lab_enc.fit_transform(y)
        x_train_data, x_test_data, y_train_data, y_test_data =
train_test_split(x, encoded_Y, test_size=0.33)
        learning_model = LogisticRegression(solver='lbfgs')
        learning_model.fit(x_train_data, y_train_data)
        traned_nodes = learning_model.predict(x_test_data)
        traned_nodes.sort()
        miner_pool = traned_nodes[:: -1]
        miner_node = miner_pool[0]
        return miner_node

    def cal_hash(self):
        block_data = json.dumps(self)
        return sha256(block_data.encode()).hexdigest()


    @app.route('/select_miner_and_mine', methods=['POST'])
    def use_selected_miner(self):
        miner_node = select_miner()
        miner_node_addr = request.get_json()["node_address", miner_node]
        if not miner_node_addr:
            return "Node not Found", 400
data = {"node_addr": request.host_url}
        headers = {'Content-Type': "application/json"}
        response = requests.post(miner_node_addr + "/select_miner",
                                 data=json.dumps(data), headers=headers)

        if response.status_code == 200:
            res = blockchain.mine_block()
            if not res:
                return "transaction is not mined"
            else:
                record_list_len = len(blockchain.record_list)
                do_consensus()
                if record_list_len == len(blockchain.record_list):
                    publish_new_block(blockchain.get_last_block)
                return blockchain


class Blockchain:
    diff_level = 2

    def __init__(self):
        self.pen_tx = []
        self.record_list = []

    def create_gen_block(self):
        initial_block = Block(0, [], 0, "0")
        initial_block.hash = initial_block.cal_hash()
        self.record_list.append(initial_block)

    def get_last_block(self):
        return self.record_list[-1]

    def add_nw_block(self, block_info, prof_correctness):
        preblock_hash = self.get_last_block.hash
        if preblock_hash != block_info.preblock_hash:
            return False

        if not Blockchain.eval_proof_of_work(block_info, prof_correctness):
            return False
        block_info.hash = prof_correctness
        self.record_list.append(block_info)
        return True

    def proof_of_work_consen(block_info):
        block_info.nonce_val = 2
        hash_val = block_info.cal_hash()
        while not hash_val.startswith('0' * Blockchain.diff_level):
            block_info.nonce_val = block_info.nonce_val + 1
            hash_val = block_info.cal_hash()
        return hash_val

    def add_new_tx(self, tx):
        self.pen_tx.append(tx)
    def eval_proof_of_work(cls, block_info, block_hash):
             block info = block_hash.startswith('0' * Blockchain.diff_level)
             if(block_hash == block_info.cal_hash())
                    prof_correctness = True
             else
                    prof_correctness = False
             return block_info and prof_correctness


    def check_record_list_validity(cls, record_list):
        result = True
        preblock_hash = "0"
        for block in record_list:
            block_hash = block.hash
            if not cls.eval_proof_of_work(block, block_hash) or
                    preblock_hash != block.preblock_hash:
                result = False
                break
            block.hash, preblock_hash = block_hash, block_hash
        return result

    def do_consensus():
        global blockchain
        longest_record_list = None
        current_len = len(blockchain.record_list)
        for system in peer_nodes:
            response = requests.get('{}record_list'.format(system))
            pre_len = response.json()['length']
            record_list = response.json()['record_list']
            if len > current_len and
blockrecord_list.check_record_list_validity(record_list):
                current_len = pre_len
                longest_record_list = record_list
        if longest_record_list:
            blockchain = longest_record_list
            return True
        return False

    def mine_block(self):
        if not self.pen_tx:
            return False
        last_block = self.get_last_block
        new_block = Block(n_r=last_block.n_r + 1,
                          tx_data=self.tx_data,
                          adding_time=time.time(),
                          preblock_hash=last_block.hash)
        self.pen_tx = []
        return True
```

Appendix 2 Simulator Screenshot



```
Amount: 1000$
To: Alex
From: Marina
```



Previous Block Hashcode : 86113360b605e5a0080592dfc8d05d84

Transaction : 1

Data : Amount: 1000$ To: Alex From: Marina

Timestamp *21:53*

Appendix 3 – Non-exclusive licence for reproduction and publication of a graduation thesis[1]

I Ashu Dhama

1. Grant Tallinn University of Technology free licence (non-exclusive licence) for my thesis secured IoT based on blockchain based technology, supervised by Muhidul Islam Khan
   1.1. to be reproduced for the purposes of preservation and electronic publication of the graduation thesis, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright;
   1.2. to be published via the web of Tallinn University of Technology, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright.
2. I am aware that the author also retains the rights specified in clause 1 of the non-exclusive licence.
3. I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from other legislation.

02.01.2021

[1] The non-exclusive licence is not valid during the validity of access restriction indicated in the student's application for restriction on access to the graduation thesis that has been signed by the school's dean, except in case of the university's right to reproduce the thesis for preservation purposes only. If a graduation thesis is based on the joint creative activity of two or more persons and the co-author(s) has/have not granted, by the set deadline, the student defending his/her graduation thesis consent to reproduce and publish the graduation thesis in compliance with clauses 1.1 and 1.2 of the non-exclusive licence, the non-exclusive license shall not be valid for the period.