applied sciences

MDPI

Article

Efficient Load Balancing for Blockchain-Based Healthcare System in Smart Cities

Faheem Nawaz Tareen [1], Ahmad Naseem Alvi [1], Asad Ali Malik [1], Muhammad Awais Javed [1], Muhammad Badruddin Khan [2,*], Abdul Khader Jilani Saudagar [2], Mohammed Alkhathami [2] and Mozaherul Hoque Abul Hasanat [2]

[1] Department of Electrical and Computer Engineering, COMSATS University Islamabad, Islamabad 45550, Pakistan

[2] Information Systems Department, College of Computer and Information Sciences, Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh 11432, Saudi Arabia

* Correspondence: mbkhan@imamu.edu.sa

Abstract: Smart cities are emerging rapidly due to the provisioning of comfort in the human lifestyle. The healthcare system is an important segment of the smart city. The timely delivery of critical human vital signs data to emergency health centers without delay can save human lives. Blockchain is a secure technology that provides the immutable record-keeping of data. Secure data transmission by avoiding erroneous data delivery also demands blockchain technology in healthcare systems of smart cities where patients' health history is required for their necessary treatments. The health parameter data of each patient are embedded in a separate block in blockchain technology with SHA-256-based cryptography hash values. Mining computing nodes are responsible to find a 32-bit nonce (number only used once) value for each data block to compute a valid SHA-256-based hash value in blockchain technology. Computing nonce for valid hash values is a time-taking process that may cause life losses in the healthcare system. Increasing the mining nodes reduces this delay; however, the uniform distribution of mining data blocks to these nodes by considering the priority data is a challenging task. In this work, an efficient scheme is proposed for scheduling nonce computing tasks at the mining nodes to ensure the timely execution of these tasks. The proposed scheme consists of two parts, the first one provides a load balancing scheme to distribute the nonce execution tasks among the mining nodes such that makespan is minimized and the second part prioritizes more sensitive patient data for quick execution. The results show that the proposed load balancing scheme effectively allocates data blocks in different mining nodes as compared to round-robin and greedy algorithms and computes hash values of most of the higher-risk patients' data blocks in a reduced amount of time.

Keywords: blockchain; healthcare; smart cities

check for updates

Citation: Tareen, F.N.; Alvi, A.N.; Malik, A.A.; Javed, M.A.; Khan, M.B.; Saudagar, A.K.J.; Alkhathami, M.; Abul Hasanat, M.H. Efficient Load Balancing for Blockchain-Based Healthcare System in Smart Cities. *Appl. Sci.* **2023**, *13*, 2411. https://doi.org/10.3390/app13042411

Academic Editor: Dimitris Mourtzis

Received: 1 February 2023
Accepted: 7 February 2023
Published: 13 February 2023

1. Introduction

In recent years, the concept of smart cities has emerged as a new paradigm in improving the quality of life. By employing modern information and communication technologies, smart cities are capable of catering to efficient energy management, intelligent transportation facilities, waste management, enhanced health monitoring, and and efficient resource management [1–4]. The data received from different IoT devices used in various smart city applications are stored in the cloud. Applying artificial intelligence, machine learning, and modern data analytic techniques to these cloud data are envisioned to play a vital role in enhancing the economic growth of a country by the optimal utilization of available resources [5].

With a rapidly growing number of ubiquitously connected IoT devices in different applications of smart cities, the data produced by these devices are growing exponentially [6,7]. To handle such a large amount of data, recently, fog computing has emerged as a new paradigm. Fog nodes consist of single or multiple data centers located at the edge of user networks without routing over the internet backbone [8–12]. Consequently, the sensor devices belonging to the same IoT network can transmit their data at faster rates for quick processing. Fog computing is a better choice for delay-sensitive applications, such as emergency response in multiple smart cities applications, such as healthcare services [13,14].

The healthcare system is one of the prime applications in smart cities [15–17]. In the healthcare system, the vital signs data of different humanly planted sensors are required to be examined by remotely placed medical physicians for online monitoring. In case the vital signs of the patient increase their threshold limits, necessary action is urgently required. To ensure the valid delivery of patients' data, a secure data-delivery mechanism, such as blockchain, is required. Moreover, the timely delivery of healthcare services to patients is also a critical challenge [18–20].

Blockchain, also known as a system of proofs, has been emerging rapidly in many applications over the last few years. Blockchain is a secure method of handling data that results in immutable and decentralized maintenance of records in the form of blocks. The data are stored in the blocks along with the secure hash functions. The blocks are chained together by using the data of the previous block to compute the hash function of the current block as shown in Figure 1.

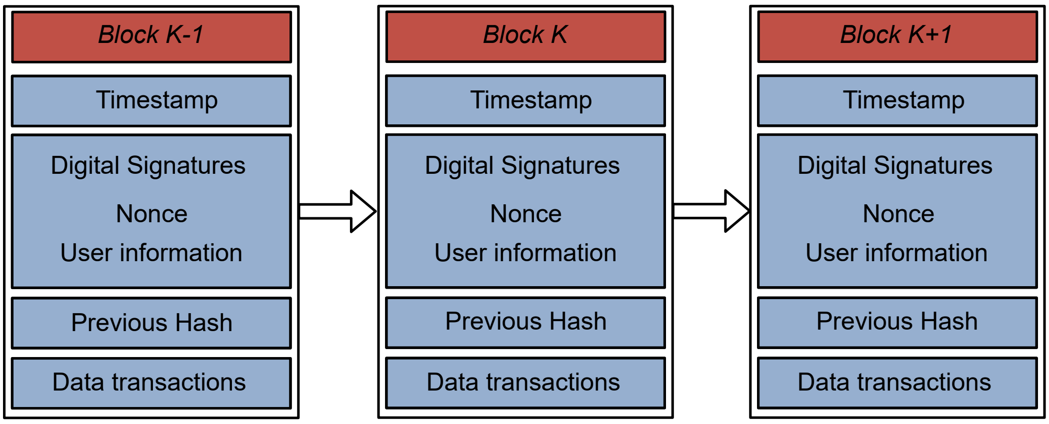

Figure 1. The structure of blockchain blocks.

Immutability is one of the key features in blockchain technology that allows blockchain ledgers to keep their time-stamped transaction data permanent and unalterable. Furthermore, immutability in blockchain provides the fast, efficient, and cost-effective audit of transaction data. This secure data transaction in blockchain attracts multiple supply chain applications, such as agri-food supply [21], medical equipment demand and supply [22], and perishable food products in the supply chain [23]. Immutability is achieved through cryptography hashes by applying the secure hash algorithm (SHA-256). Miners in blockchain technology are responsible for generating these hash values for each block in blockchain systems [24].

There are four different types of blockchain, such as public, private, hybrid, and consortium blockchain. These have been categorized as permissionless, permissioned, and both. A public blockchain is permissionless and decentralized, which allows anyone to join the blockchain network with equal rights. It is mainly used in cryptocurrency. Conversely, a private blockchain is permissioned and partially decentralized, which restricts network access to certain nodes with varying rights limitations. A hybrid blockchain is a mixture of permissionless and permissioned blockchains. It is centrally controlled by a single organization like a permissioned blockchain; however, to perform certain transaction validations, it offers a level of oversight similar to the public blockchain. A consortium blockchain is a permissioned blockchain technology and is governed by a group of organizations rather than a single entity. A consortium blockchain is more decentralized as compared to a private blockchain, resulting in a higher level of security. The healthcare systems can use the consortium blockchain to store and transfer data of the patients among different hospitals.

The evolution of wireless body area networks since the last decade has increased the delivery of patient data to health physicians in smart cities. Critical patients require emergency treatment, and their health parameters data are required to be transmitted to healthcare systems for immediate medical action. In addition, the vulnerability in wireless communication causes erroneous data delivery, which results in serious cybersecurity challenges. For proper data delivery and to avoid misleading information, a blockchain-based mechanism is proposed that allows only legitimate patients' data to be forwarded to medical advisors for timely decisions to avoid any causalities.

Blockchain supports the recording and tracking of resources. In a blockchain, a valid data block is generated by a mining computing node. The mining node computes a unique number called nonce against each SHA-256-based valid hash value for each data block. The fog computing node in smart cities acting as a mining node provides trustfulness of the data. Moreover, due to its proximity to patients' data, it provides quick data delivery to its mining node. However, computing a nonce for hashing key within a specific range consumes heavy computation and may result in delays that are not required in transferring sensitive patient data in healthcare systems.

To efficiently compute the nonce, special mining nodes with high computing capacity can be installed in the network. A major challenge in such a blockchain system is to design efficient algorithms for offloading the nonce computing tasks to the mining nodes. The offloading algorithms should not overburden a particular mining node. Thus, load balancing among the mining nodes must be ensured.

In this work, an efficient algorithm for offloading nonce computing tasks for a data block of delay sensitive patients' data for smart cities healthcare system (DSP_{SHS}) is proposed. DSP_{SHS} offers a load balancing mechanism along with two algorithms, one for the load balancing node to efficiently distribute the data blocks to mining nodes and the second for the mining nodes to prioritize the nonce computation task for high-priority patients.

The major features of DSP_{SHS} are described as follows:

1. The proposed scheme reduces the delay in nonce computation tasks by offering a load balancing mechanism and thus computes more nonce calculation tasks in a specific time. This is achieved by optimally distributing the data blocks to different mining nodes to compute their respective SHA-256-based hash values.
2. DSP_{SHS} proposes an algorithm for the load balancer to optimally distribute the nonce requesting data blocks on different mining nodes.
3. DSP_{SHS} proposes another algorithm for mining nodes to prioritize the highest-priority patient data blocks over comparatively low-priority data blocks.

The rest of the paper is organized as follows: Section 2 describes the previous research work in the blockchain and smart cities. The system model and our proposed scheme are discussed in Sections 3 and 4, respectively. Comparative analysis with the simulation results is discussed in Section 5. Section 6 concludes the manuscript.

2. Related Work

We divided the literature review section into four parts. The first subsection reviews the use of blockchain in healthcare systems. The second subsection surveys the efficient blockchain mining techniques that do not involve load balancing. The third subsection describes various load-balancing techniques for blockchain systems in the literature. Lastly, the last subsection explains the novelty of the proposed technique.

2.1. Blockchain in Healthcare

Blockchain technology is widely used in diverse applications and is in high research areas these days. Zhang et al. [25] propose a consortium blockchain model for data sharing. The proposed technique uses digital signature-based hash functions to ensure data privacy. A tolerance system based on the Byzantine technique is developed and used for blockchain transactions. The proposed work improves the data transmission capabilities of the blockchain.

Shrestha et al. [26] focus on a vehicular network environment and divide the area into different geographical regions for efficient blockchain-based data transmission. By using the efficient clustering of vehicular nodes, the data are transmitted with low latency in varying traffic density scenarios. Khan et al. in [27] introduce a different type of blockchain to increase the trust level of data transmissions in a vehicular network. In particular, techniques such as certificates, trust algorithms, and revocation schemes are proposed by the authors to improve the security of data delivery in the presence of malicious users.

Smart cities are emerging rapidly due to the provisioning of ease in human lifestyle, such as the healthcare system. Secure and trusted data delivery are key challenges in the healthcare system of smart cities. Due to secure data delivery provision in blockchain technology, its utility in healthcare systems is under high research area these days.

In [28], the distributed dynamic mutual identity authentication (DDMIA) system is proposed for such patients that are referred to specialist healthcare from their primary healthcare center. DDMIA follows blockchain technology for transferring patient data to the referred medical care without using the registration process. Dass et al. [29] emphasized the exchange of patient information by using blockchain technology and proposed a large-scale information infrastructure to access smart contracts as information mediators. These smart contracts are sponsored by Electronic Health Record, which allows immutable, authentic accessible medical health records for privacy and faster payments.

In [30,31], problems of the patient-centric healthcare system are highlighted, and a blockchain-based solution is proposed. The solution proposed in [30] is based on the uncheckable distributed ledger technology of blockchain to protect the patients' data privacy with increased security. In [31], the authors investigated the issues of the internet of medical things (IoMT), blockchain technologies, and cloud computing and proposed real-time remote healthcare of a one-to-one care structure.

Vasilatean et al. [32] proposed an IoT and blockchain-based healthcare system by considering the aging population and catering to the increasing cost of elderly patient care. The authors validated their proposed scheme with a case study and analyzed how elderly patients behave in an ambient assisted living environment. In [33], IoT-based healthcare systems are highlighted in different prospects, and the blockchain-based identity and access management (IAM) systems in the healthcare environment are discussed. The authors presented a systematic review of blockchain-based IAM systems to investigate data security, risk management, and functional parameters.

2.2. Efficient Mining of Blockchain Tasks

Several other techniques propose the efficient mining of blockchain tasks but do not involve load balancing. The work in [34] consider a blockchain-assisted mobile edge computing scenario. The goal of the work is to jointly optimize the privacy of mining computing task offloading and the profit of mining nodes. A reinforcement learning-based solution is provided to improve the system security, reduce the network costs and increase the system revenue.

In [24], the idea of using non-mining nodes to assist in the mining of blockchain tasks is presented. The problem is formulated as an auction game, and the optimal pricing for the auction is obtained. Similarly, in the work in [35], the problem of mining task offloading is considered from the perspective of various stakeholders, such as mining nodes and mobile edge computing service providers. To jointly optimize the utilities of mining nodes and service providers, a Stackelberg-based game theoretic approach is used.

The work in [36] proposed a novel utility function considering multiple factors for blockchain task offloading. The idea is to maximize the system utility based on wireless channels, computational resource allotment, and transmission power selection. The work uses a deep reinforcement learning approach to improve the system's utility.

In [37], the authors proposed a novel drone-based architecture to efficiently offload the mining tasks from IoT devices to the edge and the cloud. The work in [38] considered a blockchain-assisted mobile edge computing scenario and proposed a deep learning-based offloading algorithm to improve the task execution delay in a multi-user multi-fog node environment.

2.3. Load Balancing in Blockchain Networks

Load balancing in blockchain networks has been considered in several works where researchers have proposed a fair method of executing blockchain-related tasks. In Table 1, we present a summary of these techniques along with our proposed idea at the end.

Table 1. Load balancing techniques used in blockchain.

Blockchain Network Used	Load Balancing Technique	Key Idea
Vehicular network [39]	Auction mechanism	Clustering mechanism Transmission rate improvement Fairness
IoT [40]	Genetic algorithm Simple additive weighting Multicriteria decision making	Find feasible resource allocation multi-objective optimization
Electric vehicles [41]	Contract incentive mechanism	Incentive based energy cooperation
Smart grids [42]	P2P energy trading	Smart contract for energy trading
Financial systems [43]	Kademlia based DHT	Data sharing based on DHT
Financial systems [44]	Round Robin	Block creation performance Block validation performance
IoT (Our technique)	Minimum makespan Prioritization	Fair distribution of nonce tasks Quick execution of sensitive data

Authors in [39] considered a vehicular network scenario, where vehicles offload their mining tasks to the roadside units. The idea of the work is to divide the road area into clusters and assign each mining node to a cluster. By using an auction-based game theoretic algorithm, the work increases the transmission rate of mining tasks and also ensures the fairness of tasks at the mining nodes.

In [40], a genetic algorithm is proposed to evaluate the feasible resource allocation solution for IoT networks. After the first step, simple additive weighting and multicriteria decision making are used to find the optimal allocation of resources such that the load is balanced, and the requirements of task delay and node energy consumption are achieved.

The work in [41] considered a blockchain-based electric vehicle scenario, where energy cooperation is desired. A contract incentive mechanism is proposed to promote energy cooperation among the vehicles and achieve energy load balancing. Similarly, the work in [42] introduced a peer-to-peer energy trading mechanism for blockchain-based smart grids. Smart contracts were developed for energy trading such that electric load balancing is achieved.

In [43], a financial transaction system is considered, and an efficient data-sharing scheme is proposed so that the data load is balanced. The Kademlia-based distributed hash table (DHT) is proposed for broadcasting blockchain-related data. Additionally, the work in [44] proposed a round-robin mechanism for financial blockchains. The detailed performance analysis of such blockchain systems and the impact of load balancing is studied. The block creation and block validation performance are analyzed.

2.4. Novelty of the Proposed Technique

The current work in the literature is focused on using game theoretic techniques and machine learning algorithms for maximizing the blockchain task transmission rates and profit of the mining nodes. Moreover, load balancing is generally achieved using game theoretic techniques, where an auction or incentive is used to promote load balancing. The simple round robin load balancing techniques have also been used to improve blockchain systems.

In comparison, we consider a blockchain-assisted healthcare system model which has not been considered for load balancing. We also utilize the minimum makespan-based load balancing technique to efficiently schedule the nonce computing tasks which has not been used in the context of blockchain mining. Moreover, we also propose a priority algorithm to improve the task computation delay of high-priority patients' data.

3. System Model

The patient's healthcare data collected through multiple healthcare sensors are forwarded directly to the load balancing node. To convert this data block as a member of the blockchain, a unique nonce value to determine the hash value within a specified range is required to be computed. To efficiently compute a specific nonce value, the load balancing node forwards these data blocks to one of its directly connected fog mining nodes by applying a load balancing algorithm. Each fog mining node fetches one of the data blocks placed in its cache to compute nonce. Mining nodes compute nonce by generating random numbers to meet the specified conditions. The successful data block with a nonce value is sent back to the load balancer to become a part of the blockchain and forwarded to the healthcare center for necessary action through the internet cloud. We assume that nonce computing tasks are transmitted using traditional cryptographic schemes and the security for the transmission of these tasks is not the focus of this paper.

In this work, we considered three different severity levels of a varying number of patient data, such as high-risk patient data, moderate-risk patient data, and low-risk or routine patient data. Each of these patient data is treated separately and is required to be sent to the centrally placed healthcare system of a smart city. These patients' data are forwarded to one of the sparsely placed load balancing nodes that are backwardly connected with multiple fog mining nodes. A load-balancing node randomly receives a varying number of data requests that are required to be forwarded to one of its neighboring mining nodes by considering patient data severity and the load on the mining fog computing nodes. Each mining node has a uniform caching capacity to store a data block before computing its nonce value that meets the specific conditions.

There are Y healthcare application networks placed in different localities of a smart city. Each healthcare network Y_j comprises three different severity levels of patient data and are categorized as h_{ls}, h_{ms}, and h_{hs} as low, medium, and highly sensitive data, respectively. Each Y_j consists of X number of fog mining nodes and each X_i fog mining node has N number of blocks. If N_{ls}, N_{ms}, and N_{hs} are the total number of low, medium, and highly sensitive types of data blocks, respectively, which are received by all fog mining nodes, then they are computed in Equations (1)–(3) as

$$N_{ls} = \sum_{j=1}^{Y} \sum_{i=1}^{X} N_{ij}(h_{ls}) \tag{1}$$

$$N_{ms} = \sum_{j=1}^{Y} \sum_{i=1}^{X} N_{ij}(h_{ms}) \tag{2}$$

$$N_{hs} = \sum_{j=1}^{Y} \sum_{i=1}^{X} N_{ij}(h_{hs}) \tag{3}$$

The total number of data blocks (N_{DB}) received by all fog mining nodes placed in a smart city is calculated as:

$$N_{DB} = N_{ls} + N_{ms} + N_{hs} \tag{4}$$

The patient's data are forwarded directly from the patient to the central load-balancing node using wireless communications, such as 5G. Similarly, the data from the the load-balancing node to the mining node also use 5G communications.

We assume that there are multiple channels available between the load balancing node and healthcare patients' data nodes to transmit all required data blocks simultaneously without considering any queuing delay. Similarly, there are multiple channels between load-balancing nodes and fog-mining nodes. A complete system model is shown in Figure 2.

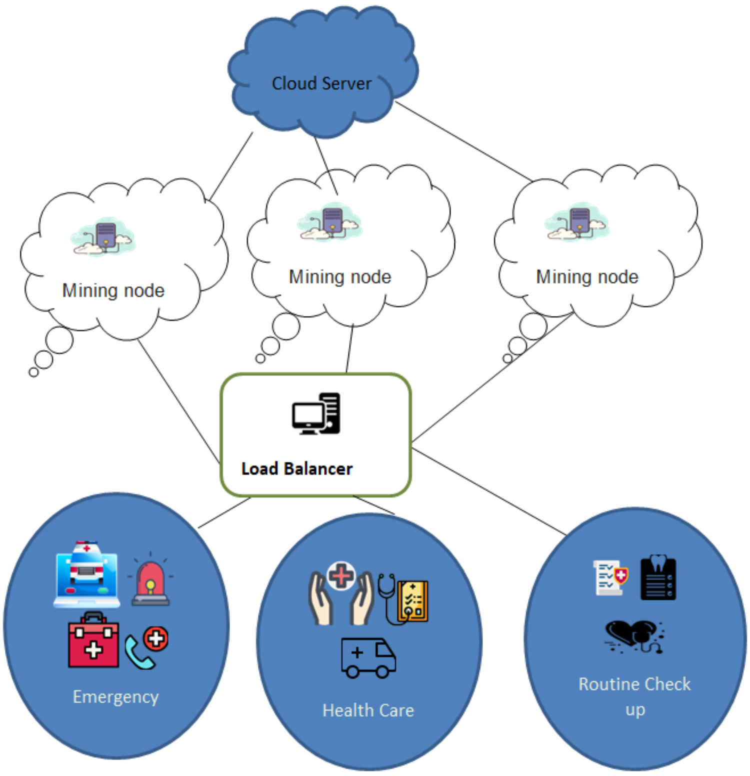

Figure 2. Load balancing in a mining node.

4. Proposed Scheme

In this work, nonce computing for a data block of delay sensitive patients' data for smart cities healthcare system (DSP_{SHS}) is proposed. DSP_{SHS} comprises two algorithms, one for the load balancer and another for the fog mining node discussed in Sections 4.1 and 4.2, respectively. Multiple healthcare data arrive at the load balancer, which are required to be forwarded to one of the directly attached mining nodes to determine their nonce value. The load balancer forwards these data blocks to mining nodes by applying algorithms described in Section 4.1. The data blocks are stored in the cache of the mining node. When there are multiple data blocks in the cache of the mining node, then it fetches one of them to compute nonce by applying the algorithm discussed in Section 4.2.

4.1. Proposed Algorithm for Load Balancing Node

In smart cities, multiple fog nodes act as mining computing nodes to find a valid hash value of a patient's health data block. Some of these mining nodes have more data blocks for execution, whereas some have no data to compute. In this work, we introduced the concept of a load balancer for mining nodes in a smart city healthcare system. The primary goal of load balancing is to ensure that no one mining node is overloaded or under-loaded.

The purpose of load balancing is to reduce the computation time. IoMT continuously sends patients' vital signs data to the mining nodes. The mining nodes have a limited processing capability. Different mining nodes are placed in different locations in the smart city. It may be possible that one mining node may be overloaded, and the other mining nodes may have lesser computational nonces of data blocks. Therefore, there is a need to have a load balancer to equally distribute the requesting data to each mining node. In this work, we proposed an algorithm for efficient load balancing in multiple mining computing nodes by applying a modified greedy algorithm.

In this section, an algorithm for the load-balancing node to fairly distribute the patients' data blocks to the mining nodes is proposed. In this algorithm, a traditional greedy algorithm is modified to distribute the data blocks to multiple mining machines to minimize the computational time.

The salient features of the proposed algorithm are as follows:

- Rearrange the entire data blocks according to their computing time relating to data block size in descending order.
- Assign the longest data block to the mining node with the lowest computing load.

Suppose we have X indistinguishable fog mining nodes, each having a data caching limit (D_{lim}) and N data blocks with varying block sizes. The size of a data block D, which is required to be cached on a fog mining node is D_j. Our main goal is to assign the data blocks to the mining node so that no fog mining node is under or overloaded. The maximum load on a fog mining node at any time instant T is calculated in two ways:

1. Calculate the total data block sizes and divide them by the number of fog mining machines to get an average of the data load that needs to be assigned on any machine. The optimum data capacity (D^1_{Opt}) in this case is determined as:

$$D^1_{Opt} \geq 1/X \times \sum_{j=1}^{N} D_j \tag{5}$$

The maximum caching load on any of the machines cannot be less than this average load. For example, if there are three data blocks D1, D2, D4, and D5 with data sizes of 2, 3, 1, 2, and 2, respectively, and they are required to be allocated on two machines, then each machine will be assigned a load of 5, which is the average of the load by allocating 3 and 2 data blocks each.

2. When there is a single large-size data block and the rest are smaller, then the large block is assigned to one fog mining machine and all the rest to other fog mining machines. In this case, the optimal load assignment on a machine (D^2_{Opt}) is calculated as

$$D^2_{Opt} \geq maxD_j \quad 1 \leq j \leq N \tag{6}$$

In this case, the maximum caching load on a single machine will be equal to the size of the biggest data block. For example, there are three data blocks D1, D2, and D3 with data sizes of 2, 1, and 6, respectively, that are required to be assigned on two machines, then one machine will be allocated D3, where D1 and D2 are allocated on other machines, resulting in a maximum caching load of 6.

In light of the above two scenarios, the optimal allocation of data (D_{Opt}) is formulated as

$$D_{lim} \geq D_{Opt} \geq max(1/X \times \sum_{j=1}^{N} D_j, maxD_j) \tag{7}$$

The goal of load balancing is to assign the data blocks on fog mining nodes in such a way that D_{Opt} should be minimized.

To achieve this goal, we modified the well-known greedy algorithm by incorporating the longest job first (LJF) algorithm to achieve the efficient distribution of load on fog mining machines. The proposed scheme is described below by providing an example.

Suppose there are 3 mining nodes called X_1, X_2, and X_3 with uniform computing capacity and 7 different sizes data blocks D1, D2, D3, D4, D5, D6, D7 with the varying load as 6, 8, 4, 10, 2, 1, and 6 respectively. The proposed algorithm allows the load balancing node to rearrange all these data blocks according to their sizes such as [“D4”, 10], [“D2”, 8] [“D1”, 6], [“D7”, 6], [“D3”, 4], [“D5”, 2], [“D6”, 1]. The proposed algorithm allows the load balancer to check the load on each mining node and assign these jobs according to the mining machine load in such a way that the data block is forwarded to that mining node that holds less load. The distribution of all these 7 data blocks on three mining nodes is shown in Figure 3.

The maximum load assigned to a mining node is 13 in this example, which is assigned to the X_1 machine. However, the load on X_2 and X_3 are 12 each.

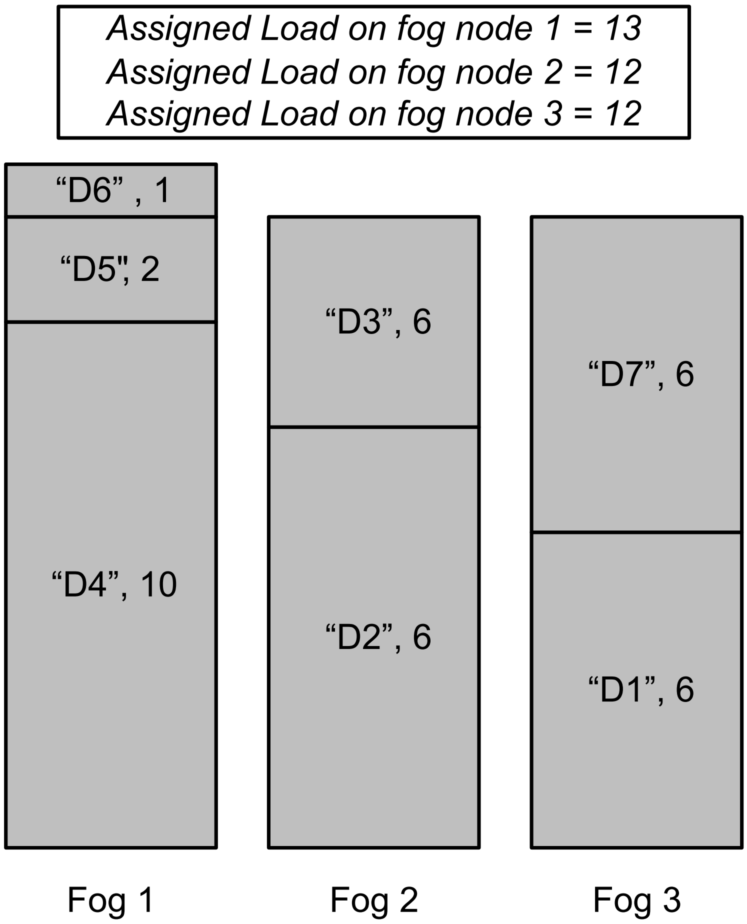

Figure 3. Example of the proposed algorithm technique.

A complete algorithm for load distribution is shown in Algorithm 1.

Algorithm 1: Proposed algorithm for load balancing

Task execution policy
Input $Datablocks = N$
Jobs= job sizes $(D1, D2, D3 \ldots Dn)$, number of fog mining nodes $(X_1, X_2, X_3 \ldots X_n)$

$Load(Xi) =$ Current load on Xi

Sort the data blocks in descending order according to their job sizes

for $i = 1$ *to* N **do**
    Allocate Di to the fog mining node j with minimum load such that its D_{lim} is not exceeded
    $Load(Xj) = Load(Xj) + Di$
end

4.2. Nonce Computing Algorithm

The data blocks received by mining nodes are initially placed in their cache before computing the nonce value. These data blocks may have different priorities depending on the patients' critical vital signs values. Such data blocks are required to be computed on a priority basis. The second algorithm of DSP_{SHS} is proposed for the mining node to compute the nonce value of a data block according to their data priority. When data arrive at the mining node from the load balancing node, then the mining node needs to compute the data according to their sensitivity level.

Suppose the load balancer receives five data blocks DB1, DB2, DB3, DB4, and DB5, and waits for their nonce values at 12, 15, 20, 25, and 30 s, respectively. These data blocks have three different priority levels, such as H, M, and L, representing high, medium, and low priority with priority values of 300, 70, and 10, respectively. Their corresponding priority values are calculated according to Table 2.

Table 2. Priority values of data blocks.

Data Blocks	*Priority Levels*	*Time Since Arrival*	*Priority Value (P_V)*
DB1	*High*	*12*	*3600*
DB2	*High*	*15*	*4500*
DB3	*Medium*	*20*	*1400*
DB4	*Medium*	*25*	*1750*
DB5	*Low*	*30*	*300*

A mining node before fetching the data block in computing their nonce value verifies the current value of the data block. Value of each block increments after each interval of time. If there are two different priority blocks available, then their value will be computed with their time of origin. If there are blocks with the same values, then the data block with a higher priority level is selected for nonce computation. If two or more data blocks have the same value with similar priority levels, then that data block is selected, which arrives in the cache of the mining node.

A complete nonce computing algorithm for the mining node is shown in Algorithm 2. It should be noted that the load balancer node handles N data blocks at a time and distributes them to the fog mining nodes as per the proposed algorithms.

Algorithm 2: Pseudo code of nonce computing by mining node

```
Data block selection policy
Input Number of data blocks = N
  timestamp of ith data block = t_i
  Priority level of ith data block = P_i
  Value of ith data block = V_i
  for i = 1 to N do
    V_i = t_i × P_i
    //Select data block with maximum V_i
    If value and priority levels of two data blocks are same
    // Process data block that arrives first in cache
    End If
    If value of two data blocks is same and priority levels are different
    //Process block with higher value of P
    End If
end
```

5. Results and Analysis

To analyze the performance of our proposed scheme in different prospects, we created a simulation environment by taking three different sensitivity levels of multiple data blocks in the healthcare system of the smart city. The data block sizes with different sensitivity levels ranged from 300 to 900 kbytes for the highest emergency data block, 300 to 900 kbytes for a moderate healthcare data block, and 300 to 900 kbytes for routine patient data blocks with the least priority level. The nonce values of these data blocks are executed from fog mining nodes and are downloaded at the data rate of 8 Mbps. However, the downloading data rate from the cloud mining node is taken as 2 Mbps. A fog mining node's simultaneous nonce computational capability is 2 nonce/s, whereas the cloud mining server can compute 4 nonce/s. A list of simulation parameters is shown in Table 3. We compared our proposed technique with the round robin load balancing mechanism [44].

Table 3. Simulation parameters.

Parameter	Value
Fog Mining Node coverage area	1500 m
Distance between fog mining node and Nonce requesting data blocks	50–1500
Number of simultaneous data blocks for each mining node	6–30
Number of mining nodes	2–10
Downloading data rate from cloud mining server	2 Mbps
Downloading data rate from fog mining node	8 Mbps
Number of Nonce computing by fog mining node	2
Emergency data blocks	2–10
Medium level sensitive data blocks	2–10
Least sensitive data blocks	2–10

5.1. Load Balancing Greedy vs. Proposed Algorithm vs. Round Robin vs. Round Robin with LJF

Figure 4 shows the comparative results of the proposed scheme as compared to its competitors in computing nonce value for the varying number of data blocks. The number of data blocks is incremented by two data blocks from 2 to 22 data blocks simultaneously with 3 fog mining nodes. The results show that our proposed scheme outperformed compared to greedy [45], round-robin [46], and round-robin with LJF. The results show that our proposed scheme reduces the nonce computing time by 35.5%, 18.38%, and 15.94% against round-robin, greedy, and round-robin with the LJF algorithms, respectively.

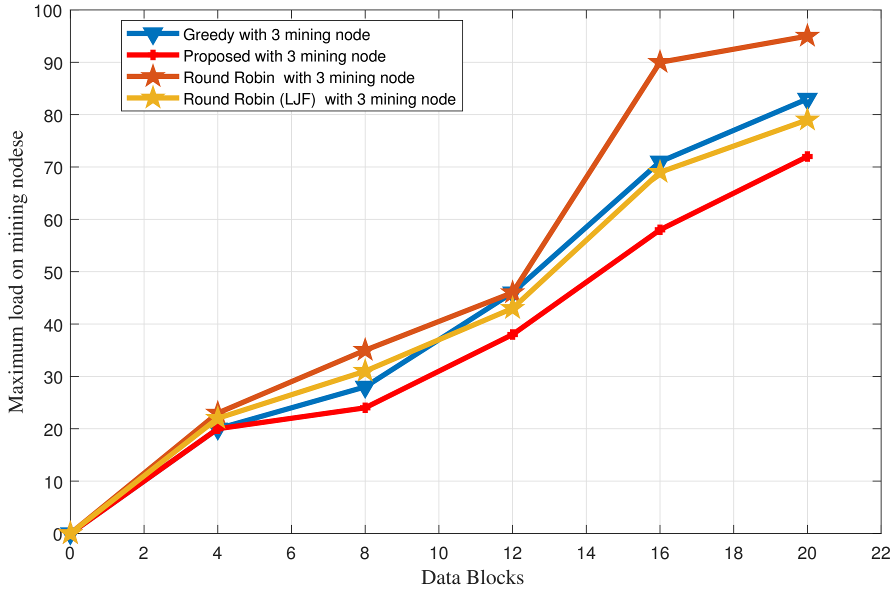

Figure 4. Number of computed nonce against a varying number of data block requests.

The results in Figure 5 show the performance of our proposed scheme in computing the nonce value against varying numbers of fog mining nodes when data blocks are fixed. It is evident from the results that total processing time reduces if we increase the no. of the fog mining node for a fixed number of 24 jobs with varying processing times. The results verify that with the increase in the number of mining nodes, the job processing time reduces because the same number of jobs are being distributed among multiple mining nodes. It is evident from the results that our proposed scheme reduces the processing time by up to 28 percent against the round-robin algorithm, up to 29.41 percent against the greedy algorithm, and up to 20 percent against the round-robin with the LJF algorithm.

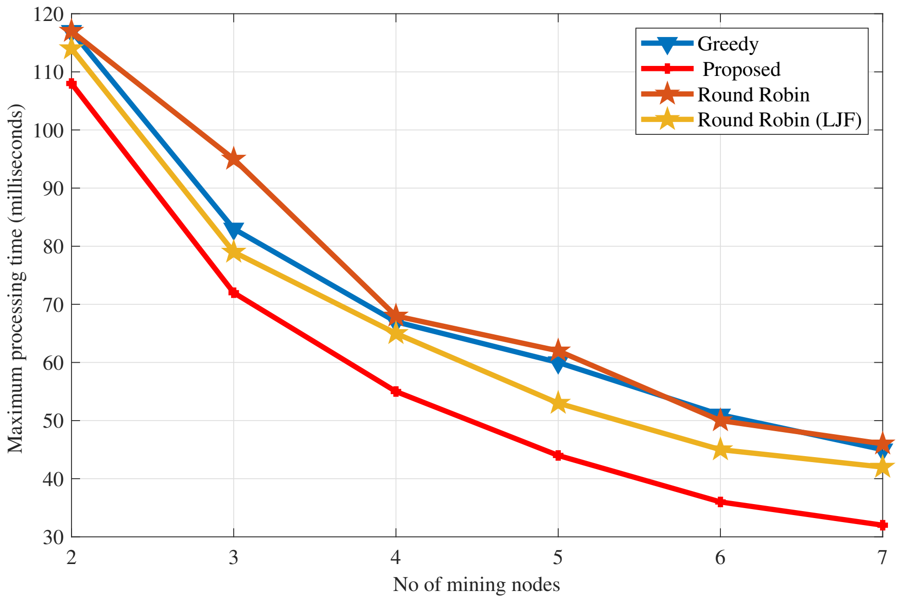

Figure 5. Nonce computing against varying number of mining nodes.

To evaluate the effectiveness of our proposed scheme, the results in terms of standard deviation ω for varying numbers of mining nodes and the varying number of jobs are shown in Figures 6 and 7, respectively.

Standard deviation is calculated as

$$\sigma = \sqrt{\sum_{i=1}^{N} (X - Y)^2 / N} \tag{8}$$

where X is the mean time required to compute the nonce value of the requested blocks, Y is the time computed by an algorithm, and N is the number of instances chosen.

Results shown in Figures 6 and 7 verify that the proposed scheme merely deviates from the mean load computed against a varying number of blocks with fixed mining nodes and for a varying number of mining nodes with fixed data blocks, respectively.

Figure 6 shows that the standard deviation values of DSP_{SHS} are far less than the other algorithms when the number of data block requests increases from 4 to 20 in five different instances and the number of fog mining nodes is 3. It is evident from the results that DSP_{SHS} merely deviates from the mean mining fog nodes load as compared to round robin, greedy, and round robin with LJF algorithms.

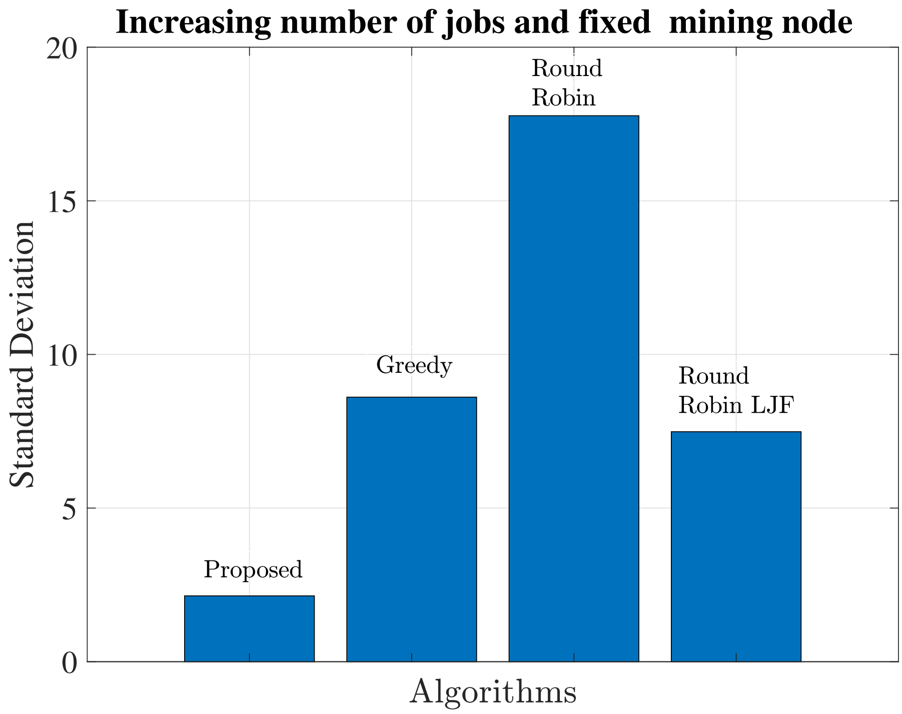

Figure 6. Standard deviation against a varying number of mining nodes.

Results in Figure 7 compute the standard deviation values for a fixed number of data blocks with varying numbers of fog mining nodes. In these results, the data block requests are fixed to 20; however, fog mining nodes are increasing from 2 to 7 in six different instances. It is evident from the results that our proposed DSP_{SHS} is very close to the mean load values of the mining nodes and prominently less than the other three schemes.

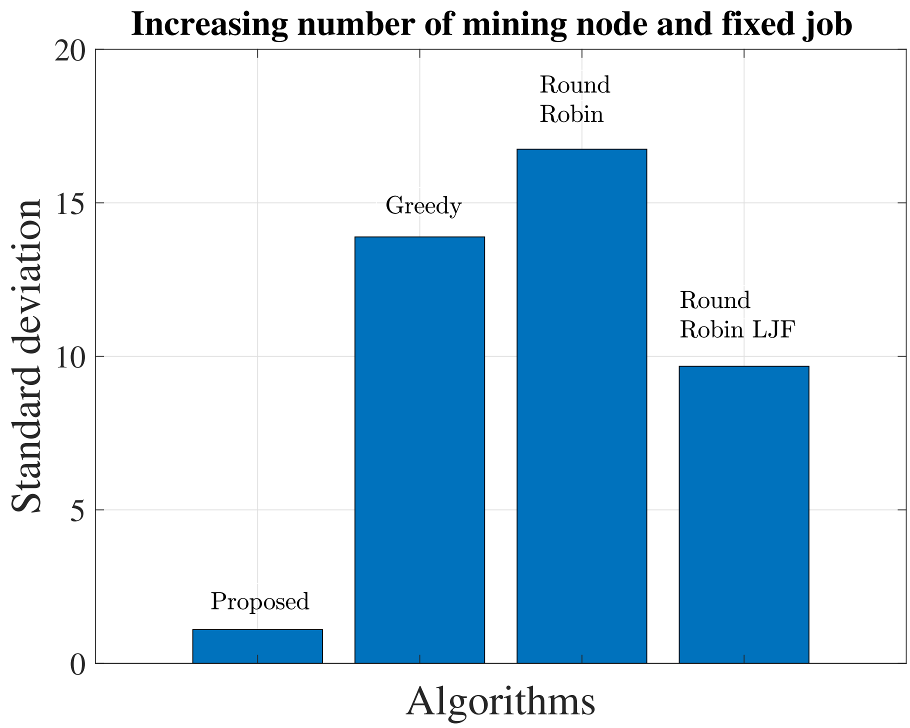

Figure 7. Standard deviation against a varying number of jobs.

5.2. Proposed Job Sequencing with the Deadline

After the efficient distribution of data blocks to mining nodes for nonce computation, the second algorithm of our proposed scheme effectively scrutinizes them in accordance with their sensitivity levels. Varying sizes of patients' data blocks are divided into three different sensitivity levels. The number of nonces against varying numbers of data blocks is such that data requests of all three sensitivity levels of data blocks are uniform. The results are compared with the first come first serve (FCFS) algorithm by considering the same computing capacity of all fog mining nodes.

The percentage of under-processed nonce computation against a varying number of data block requests for six and ten mining nodes, respectively, are shown in Figures 8 and 9, respectively. Both results include three sub-plots that represent three different priority levels data blocks, such as high-level sensitivity, medium-level sensitivity, and low sensitivity level patients' data blocks. In these results, we consider that there is an equal number of each type of data block present in the cache of each mining node, such as 2 data blocks in high-priority nonce computation data blocks indicating that there are 6 data blocks with 2 each in high, medium and low priority levels. The results shown in Figure 8 show that our proposed algorithm selects 100% nonce against high-level sensitive data blocks for varying numbers of data block requests when the number of high-priority data blocks in a mining node is less than data blocks. For medium-level sensitive data blocks, the proposed scheme computes all nonce when the number of requests is 2 each. However, when the number of nonce requests increases, the number of nonces computed by FCFS is more than our proposed scheme. This is because the mining node has already selected high-priority blocks, and it cannot process the other priority-level blocks. The same impact is shown for low-priority data blocks. It is evident from the results shown in Figure 9 that the proposed scheme accommodates most of the high-priority data blocks as compared to FCFS and only computes nonces for the medium- or low-priority blocks when there is no high-priority block left for computation.

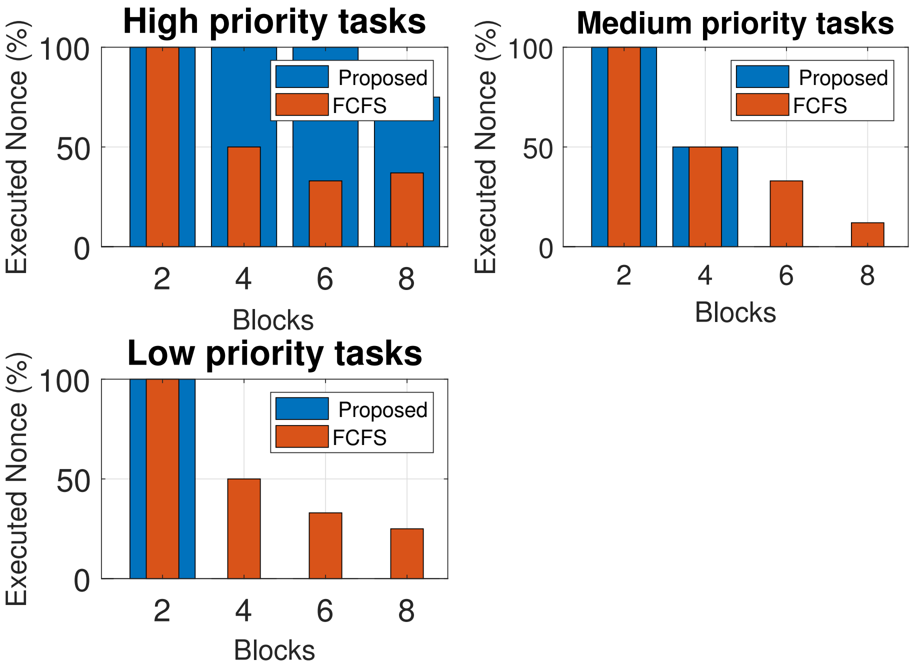

Figure 8. Percentage of the computed nonce for 6 mining nodes with varying number of data blocks.

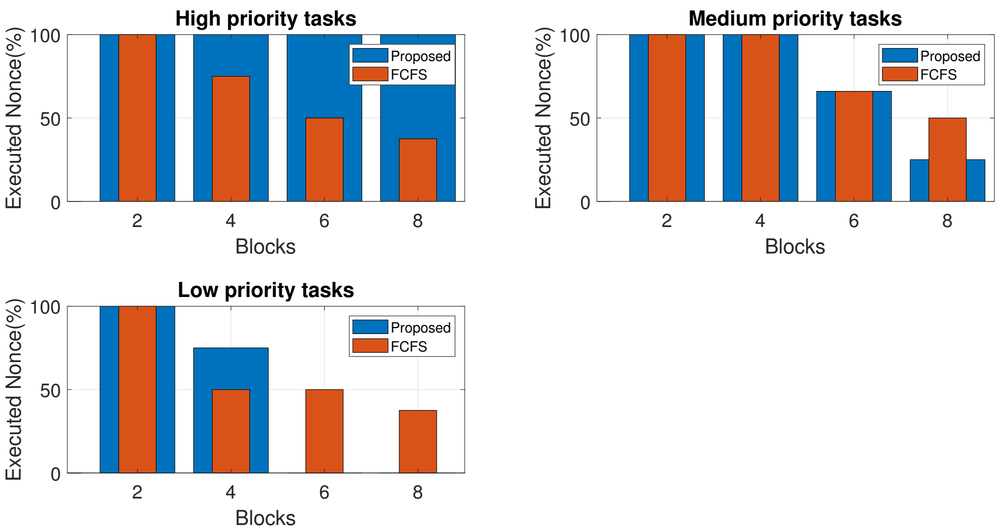

Figure 9. Percentage of the computed nonce for 10 mining nodes with varying numbers of data blocks.

Results shown in Figure 10 are obtained against varying numbers of mining nodes when there is a fixed number of nonce requests. In these results, there is a total of 12 nonce computation requests that are equally distributed in high-, medium-, and low-priority patient data blocks. It is evident from the results that our proposed scheme entertains a maximum of the high-priority data blocks and 50% of high-priority data blocks are entertained when the number of mining nodes is only two. The nonce computation of the high-priority patients' data block is 100% when the number of mining nodes is 4 and above. However, FCFS only computes the nonce of those data blocks that arrive in their cache first. The results further show that when the number of mining nodes is more than the available high-priority data blocks, then it allows medium-priority data blocks in computing their nonce values, and nonce against low-priority data blocks are only entertained when all the high-priority data blocks are successfully entertained.

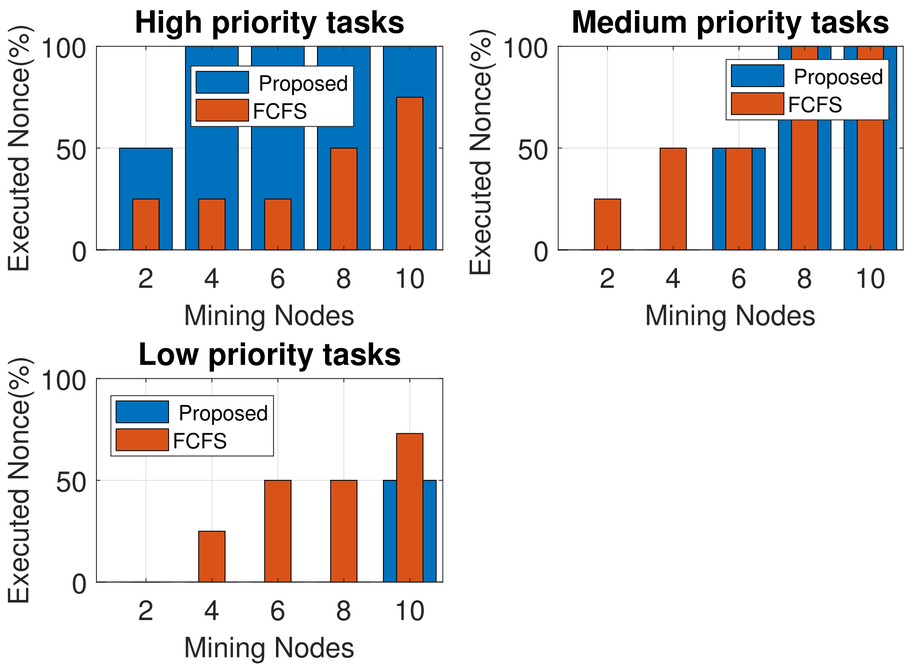

Figure 10. Percentage of the computed nonce for 12 data blocks with varying numbers of mining nodes.

6. Conclusions

The efficient computation of nonces is a major challenge in blockchain networks. This work proposes a reduced nonce computational time scheme for patients' data for smart cities healthcare system (DSP_{SHS}). In this scheme, we propose two algorithms, one for the load balancing node to equally distribute the nonce computing tasks and another one for the mining computing node to scrutinize and prioritize nonce computation for sensitive patients' data. The performance of the load balancing algorithm is compared with round robin, round robin with longest job first algorithm, and greedy algorithms. The performance of the nonce computing algorithm is compared with FCFS. It is evident from the results that (DSP_{SHS}) performs better than its competitors by computing more nonce values for data blocks of serious patients' data. In addition, it fairly distributes the data blocks in different mining nodes as compared to other algorithms. In the future, we aim to jointly optimize the latency and energy of mining nodes in blockchain networks.

Author Contributions: This article was prepared through the collective efforts of all the authors. Conceptualization, F.N.T., A.N.A., A.A.M., M.A.J., M.B.K., A.K.J.S., M.A. and M.H.A.H.; Writing—original draft, F.N.T., A.N.A., A.A.M. and M.A.J.; Writing—review and editing, M.B.K., A.K.J.S. and M.A. All authors have read and agreed to the published version of the manuscript.

Funding: The authors extend their appreciation to the Deanship of Scientific Research at Imam Mohammad Ibn Saud Islamic University for funding this work through Research Group no. RG-21-07-07.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is available from corresponding author upon request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zeadally, S.; Javed, M.A.; Hamida, E.B. Vehicular Communications for ITS: Standardization and Challenges. *IEEE Commun. Stand. Mag.* **2020**, *4*, 11–17. [CrossRef]
2. Esposito, C.; Castiglione, A.; Frattini, F.; Cinque, M.; Yang, Y.; Choo, K.K.R. On Data Sovereignty in Cloud-Based Computation Offloading for Smart Cities Applications. *IEEE Internet Things J.* **2019**, *6*, 4521–4535. [CrossRef]

3. Javed, M.A.; Nguyen, T.N.; Mirza, J.; Ahmed, J.; Ali, B. Reliable Communications for Cybertwin driven 6G IoVs using Intelligent Reflecting Surfaces. *IEEE Trans. Ind. Inform.* **2022**, *18*, 7454–7462. [CrossRef]
4. Du, R.; Santi, P.; Xiao, M.; Vasilakos, A.V.; Fischione, C. The Sensable City: A Survey on the Deployment and Management for Smart City Monitoring. *IEEE Commun. Surv. Tutorials* **2019**, *21*, 1533–1560. [CrossRef]
5. Javed, M.A.; Zeadally, S.; Hamida, E.B. Data analytics for Cooperative Intelligent Transport Systems. *Veh. Commun.* **2019**, *15*, 63–72. [CrossRef]
6. Malik, U.M.; Javed, M.A.; Zeadally, S.; Islam, S.U. Energy-Efficient Fog Computing for 6G-Enabled Massive IoT: Recent Trends and Future Opportunities. *IEEE Internet Things J.* **2022**, *9*, 14572–14594. [CrossRef]
7. Malik, U.M.; Javed, M.A. Ambient Intelligence assisted fog computing for industrial IoT applications. *Comput. Commun.* **2022**, *196*, 117–128. [CrossRef]
8. Rahim, M.; Ali, S.; Alvi, A.N.; Javed, M.A.; Imran, M.; Azad, M.A.; Chen, D. An intelligent content caching protocol for connected vehicles. *Emerg. Telecommun. Technol.* **2021**, *32*, 1–14. [CrossRef]
9. Martinez, I.; Hafid, A.S.; Jarray, A. Design, Resource Management, and Evaluation of Fog Computing Systems: A Survey. *IEEE Internet Things J.* **2021**, *8*, 2494–2516. [CrossRef]
10. Rahim, M.; Javed, M.A.; Alvi, A.N.; Imran, M. An efficient caching policy for content retrieval in autonomous connected vehicles. *Transp. Res. Part A Policy Pract.* **2020**, *140*, 142–152. [CrossRef]
11. Ali, B.; Adeel Pasha, M.; Islam, S.u.; Song, H.; Buyya, R. A Volunteer-Supported Fog Computing Environment for Delay-Sensitive IoT Applications. *IEEE Internet Things J.* **2021**, *8*, 3822–3830. [CrossRef]
12. Malik, U.M.; Javed, M.A.; Frnda, J.; Rozhon, J.; Khan, W.U. Efficient Matching-Based Parallel Task Offloading in IoT Networks. *Sensors* **2022**, *22*, 6906. [CrossRef]
13. Reis, J.; Marques, P.A.; Marques, P.C. Where Are Smart Cities Heading? A Meta-Review and Guidelines for Future Research. *Appl. Sci.* **2022**, *12*, 8328. [CrossRef]
14. Esashika, D.; Masieroa, G.; Maugerb, Y. An investigation into the elusive concept of smart cities: A systematic review and meta-synthesis. *Technol. Anal. Strateg. Manag.* **2021**, *33*, 957–969. [CrossRef]
15. Zhang, L.; You, W.; Mu, Y. Secure Outsourced Attribute-Based Sharing Framework for Lightweight Devices in Smart Health Systems. *IEEE Trans. Serv. Comput.* **2022**, *15*, 3019–3030. [CrossRef]
16. Venkatesh, J.; Aksanli, B.; Chan, C.S.; Akyurek, A.S.; Rosing, T.S. Modular and Personalized Smart Health Application Design in a Smart City Environment. *IEEE Internet Things J.* **2018**, *5*, 614–623. [CrossRef]
17. Ahmed, J.; Nguyen, T.N.; Ali, B.; Javed, A.; Mirza, J. On the Physical Layer Security of Federated Learning based IoMT Networks. *IEEE J. Biomed. Health Inform.* **2023**, *27*, 691–697. [CrossRef] [PubMed]
18. Corsini, R.R.; Costa, A.; Fichera, S.; Pluchino, A.; Parrinello, V. System design of outpatient chemotherapy oncology departments through simulation and design of experiments. *Int. J. Manag. Sci. Eng. Manag.* **2022**, 1–14. [CrossRef]
19. Edwards, L.; Hermis, K.; LeGette, C.R.; Lujan, L.A. Acuity-Based Scheduling: Outcomes in Ambulatory Oncology Centers. *Clin. J. Oncol. Nurs.* **2017**, *21*, 250–253. [CrossRef] [PubMed]
20. Kallen, M.A.; Terrell, J.A.; Lewis-Patterson, P.; Hwang, J.P. Improving Wait Time for Chemotherapy in an Outpatient Clinic at a Comprehensive Cancer Center. *JCO Oncol. Pract.* **2012**, *8*, e1–e7. [CrossRef] [PubMed]
21. Menon, S.; Jain, K. Blockchain Technology for Transparency in Agri-Food Supply Chain: Use Cases, Limitations, and Future Directions. *IEEE Trans. Eng. Manag.* **2021**, 1–15. [CrossRef]
22. Ahmad, R.W.; Salah, K.; Jayaraman, R.; Yaqoob, I.; Omar, M.; Ellahham, S. Blockchain-Based Forward Supply Chain and Waste Management for COVID-19 Medical Equipment and Supplies. *IEEE Access* **2021**, *9*, 44905–44927. [CrossRef] [PubMed]
23. Ahamed, N.N.; Thivakaran, T.K.; Karthikeyan, P. Perishable Food Products Contains Safe in Cold Supply Chain Management Using Blockchain Technology. In Proceedings of the 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 19–20 March 2021; Volume 1, pp. 167–172. [CrossRef]
24. Guo, S.; Dai, Y.; Guo, S.; Qiu, X.; Qi, F. Blockchain Meets Edge Computing: Stackelberg Game and Double Auction Based Task Offloading for Mobile Blockchain. *IEEE Trans. Veh. Technol.* **2020**, *69*, 5549–5561. [CrossRef]
25. Zhang, X.; Chen, X. Data Security Sharing and Storage Based on a Consortium Blockchain in a Vehicular Ad-hoc Network. *IEEE Access* **2019**, *7*, 58241–58254. [CrossRef]
26. Shrestha, R.; Nam, S.Y. Regional Blockchain for Vehicular Networks to Prevent 51% Attacks. *IEEE Access* **2019**, *7*, 95033–95045. [CrossRef]
27. Khan, A.S.; Balan, K.; Javed, Y.; Tarmizi, S.; Abdullah, J. Secure Trust-Based Blockchain Architecture to Prevent Attacks in VANET. *Sensors* **2019**, *19*, 4954. [CrossRef]
28. Hegde, M.; Rao, R.R.; Nikhil, B.M. DDMIA: Distributed Dynamic Mutual Identity Authentication for Referrals in Blockchain-Based Health Care Networks. *IEEE Access* **2022**, *10*, 78557–78575. [CrossRef]
29. Vardhini, B.; Dass, S.N.; R, S.; Chinnaiyan, R. A Blockchain based Electronic Medical Health Records Framework using Smart Contracts. In Proceedings of the 2021 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 27–29 January 2021; pp. 1–4. [CrossRef]
30. Zhuang, Y.; Sheets, L.R.; Chen, Y.W.; Shae, Z.Y.; Tsai, J.J.; Shyu, C.R. A Patient-Centric Health Information Exchange Framework Using Blockchain Technology. *IEEE J. Biomed. Health Inform.* **2020**, *24*, 2169–2176. [CrossRef]

31. Indumathi, J.; Shankar, A.; Ghalib, M.R.; Gitanjali, J.; Hua, Q.; Wen, Z.; Qi, X. Block Chain Based Internet of Medical Things for Uninterrupted, Ubiquitous, User-Friendly, Unflappable, Unblemished, Unlimited Health Care Services (BC IoMT U6 HCS). *IEEE Access* **2020**, *8*, 216856–216872. [CrossRef]
32. Vasilateanu, A.; Diaconu, A.T. Ambient Assisted Living Environment based on Blockchain for Elderly Care. In Proceedings of the 2020 International Conference on e-Health and Bioengineering (EHB), Iasi, Romania, 29–30 October 2020; pp. 1–4. [CrossRef]
33. Alamri, B.; Crowley, K.; Richardson, I. Blockchain-Based Identity Management Systems in Health IoT: A Systematic Review. *IEEE Access* **2022**, *10*, 59612–59629. [CrossRef]
34. Nguyen, D.C.; Pathirana, P.N.; Ding, M.; Seneviratne, A. Privacy-Preserved Task Offloading in Mobile Blockchain With Deep Reinforcement Learning. *IEEE Trans. Netw. Serv. Manag.* **2020**, *17*, 2536–2549. [CrossRef]
35. Zhang, K.; Cao, J.; Leng, S.; Shao, C.; Zhang, Y. Mining Task Offloading in Mobile Edge Computing Empowered Blockchain. In Proceedings of the 2019 IEEE International Conference on Smart Internet of Things (SmartIoT), Tianjin, China, 9–11 August 2019; pp. 234–239. [CrossRef]
36. Nguyen, D.; Ding, M.; Pathirana, P.; Seneviratne, A.; Li, J.; Poor, V. Cooperative Task Offloading and Block Mining in Blockchain-based Edge Computing with Multi-agent Deep Reinforcement Learning. *IEEE Trans. Mob. Comput.* **2021**. [CrossRef]
37. Luo, S.; Li, H.; Wen, Z.; Qian, B.; Morgan, G.; Longo, A.; Rana, O.; Ranjan, R. Blockchain-Based Task Offloading in Drone-Aided Mobile Edge Computing. *IEEE Netw.* **2021**, *35*, 124–129. [CrossRef]
38. Samy, A.; Elgendy, I.A.; Yu, H.; Zhang, W.; Zhang, H. Secure Task Offloading in Blockchain-Enabled Mobile Edge Computing with Deep Reinforcement Learning. *IEEE Trans. Netw. Serv. Manag.* **2022**, *19*, 4872–4887. [CrossRef]
39. Jameel, F.; Javed, M.A.; Zeadally, S.; Jäntti, R. Efficient Mining Cluster Selection for Blockchain-Based Cellular V2X Communications. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 4064–4072. [CrossRef]
40. Xu, X.; Zhang, X.; Gao, H.; Xue, Y.; Qi, L.; Dou, W. BeCome: Blockchain-Enabled Computation Offloading for IoT in Mobile Edge Computing. *IEEE Trans. Ind. Inform.* **2020**, *16*, 4187–4195. [CrossRef]
41. Yahaya, A.S.; Javaid, N.; Javed, M.U.; Shafiq, M.; Khan, W.Z.; Aalsalem, M.Y. Blockchain-Based Energy Trading and Load Balancing Using Contract Theory and Reputation in a Smart Community. *IEEE Access* **2020**, *8*, 222168–222186. [CrossRef]
42. Khalid, R.; Javaid, N.; Almogren, A.; Javed, M.U.; Javaid, S.; Zuair, M. A Blockchain-Based Load Balancing in Decentralized Hybrid P2P Energy Trading Market in Smart Grid. *IEEE Access* **2020**, *8*, 47047–47062. [CrossRef]
43. Kaneko, Y.; Asaka, T. DHT Clustering for Load Balancing Considering Blockchain Data Size. In Proceedings of the 2018 Sixth International Symposium on Computing and Networking Workshops (CANDARW), Takayama, Japan, 27–30 November 2018; pp. 71–74. [CrossRef]
44. Fan, C.; Lin, C.; Khazaei, H.; Musilek, P. Performance Analysis of Hyperledger Besu in Private Blockchain. In Proceedings of the 2022 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPS), Newark, CA, USA, 15–18 August 2022; pp. 64–73. [CrossRef]
45. Mukherjee, M.; Liu, Y.; Lloret, J.; Guo, L.; Matam, R.; Aazam, M. Transmission and Latency-Aware Load Balancing for Fog Radio Access Networks. In Proceedings of the 2018 IEEE Global Communications Conference (GLOBECOM), Abu Dhabi, United Arab Emirates, 9–13 December 2018; pp. 1–6. [CrossRef]
46. Ahmad, N.; Javaid, N.; Mehmood, M.; Hayat, M.; Ullah, A.; Khan, H.A. Fog-Cloud Based Platform for Utilization of Resources Using Load Balancing Technique. In *Advances in Network-Based Information Systems: The 21st International Conference on Network-Based Information Systems*; Barolli, L., Kryvinska, N., Enokido, T., Takizawa, M., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 554–567.