applied sciences

MDPI

Article

A Modified and Effective Blockchain Model for E-Healthcare Systems

Basem Assiri

Computer Science Department, Jazan University, Jazan 82917, Saudi Arabia;
bas0911@hotmail.com or babumussmar@jazanu.edu.sa

Abstract: The development of e-healthcare systems requires the application of advanced technologies, such as blockchain technology. The main challenge of applying blockchain technology to e-healthcare is to handle the impact of the delay that results from blockchain procedures during the communication and voting phases. The impacts of latency in blockchains negatively influence systems' efficiency, performance, real-time processing, and quality of service. Therefore, this work proposes a modified model of a blockchain that allows delays to be avoided in critical situations in healthcare. Firstly, this work analyzes the specifications of healthcare data and processes to study and classify healthcare transactions according to their nature and sensitivity. Secondly, it introduces the concept of a fair-proof-of-stake consensus protocol for block creation and correctness procedures rather than famous ones such as proof-of-work or proof-of-stake. Thirdly, the work presents a simplified procedure for block verification, where it classifies transactions into three categories according to the time period limit and trustworthiness level. Consequently, there are three kinds of blocks, since every category is stored in a specific kind of block. The ideas of time period limits and trustworthiness fit with critical healthcare situations and the authority levels in healthcare systems. Therefore, we reduce the validation process of the trusted blocks and transactions. All proposed modifications help to reduce computational costs, speed up processing times, and enhance security and privacy. The experimental results show that the total execution time using a modified blockchain is reduced by about 49% compared to traditional blockchain models. Additionally, the number of messages using modified blockchain is reduced by about 53% compared to the traditional blockchain model.

Keywords: parallelism; distributed systems; modified blockchain technology; personal health records; e-healthcare

check for updates

Citation: Assiri, B. A Modified and Effective Blockchain Model for E-Healthcare Systems. *Appl. Sci.* **2023**, *13*, 12630. https://doi.org/10.3390/app132312630

Academic Editor: Gianluca Lax

Received: 25 October 2023
Revised: 21 November 2023
Accepted: 22 November 2023
Published: 23 November 2023

1. Introduction

Technological development plays vital roles in areas of life such as healthcare, education, tourism, national security, and others. In the field of healthcare, healthcare agencies compete through applying advanced technologies to improve their throughput, management, control, and services with reduced costs. One important step toward this direction is to use electronic personal health records (EPHRs), which supports the use of other technologies [1]. The use of EPHRs involves cloud storage, which allows for more control, availability, and accessibility [2,3]. However, having EPHRs in cloud storage is called a centralized parallel and distributed system, which has a single point of failure.

Blockchain technology is one kind of distributed system that runs in a decentralized manner [4], in which multiple transactions are processed and grouped into one block. The blocks are listed in one ledger. Copies of the ledger are distributed among all nodes, such that every node has an updated copy of the ledger. The nodes are devices that belong to the blockchain network, and they are authorized to store and validate transactions and blocks. Actually, transactions are executed by users, but they cannot confirm (commit) those transactions. The blockchain nodes (miners) perform processing of transactions to confirm them. During this process, the miners compete in transaction processing to create a block of transactions using some consensus mechanisms, such as proof-of-work (PoW) or proof-of-stake (PoS) [5]. Then, the miner who succeeds in creating a new block proposes that block to other miners in order to verify it. If this block is verified, then it is added to the ledger; otherwise, it is ignored [6,7]. The blockchain processes are executed as follows:

- Proposal: A miner verifies transactions and proposes a new block to other miners (they will act as validators);
- Verification: The validators validate the proposed block and send their votes to the others to either confirm or decline (commit or abort) the proposed block;
- Consensus: After receiving the votes of all validators, every validator checks the votes of the majority. Accordingly, the block is committed and either added to the ledger or not.

On the other hand, healthcare utilizes various data sources, such as healthcare professionals and the Internet of Things (IoT). Firstly, healthcare professionals have different levels of authority and trustworthiness, and this is also connected with the criticality of healthcare situations. For example, in emergency cases, doctors and nurses should access EPHRs to read or update them directly without any delay; for these purposes, blockchain procedures such as permission or voting cannot be applied. Secondly, IoT, including sensors, smartphones, and wearable mobile devices, provides real-time data as these devices sense and reflect data directly [8]. These devices are able to facilitate or perform some actions [9–11]. Actually, wearable mobile devices can be embedded in clothes or accessories such as watches, bracelets, glasses, jewelry, etc. [12]. There are also other, complicated kinds of wearable devices that can be embedded into the human body. This allows for the improvement of healthcare follow-up and services. It helps in tracking life signs and monitoring patients' situations [13]. However, such real-time technology is challenged by delays caused by blockchain procedures [14]. Moreover, these tools usually have limited storing, processing, and energy capabilities, which would also be challenged by blockchain procedures such as mining, validating, voting and storing processes [15].

Applying blockchain technology to e-healthcare has many advantages, such as decentralization, security, privacy, anonymity, transparency, reliability, and fault tolerance. However, the main challenge is to handle the impact of the latency that results from blockchain procedures during the communication and voting phases. The impacts of latency in blockchains negatively influence systems' efficiency, performance, real-time processing, and quality of services. To the best of our knowledge, this is the first work that modifies the blockchain model to cope with e-healthcare authority levels and real-time specifications.

This work proposes a modified model of a blockchain that is used to store, process, and manage data in the field of e-healthcare. Firstly, this work studies and classifies healthcare data according to their nature and sensitivity. It investigates and analyzes the roles and authorities that are interwoven with data access and processing. Secondly, understanding the nature of transaction is an important step at the beginning of this work. Unfortunately, many works apply blockchain technology without studying the implications of using transactions, which obviously shows a lack of understanding of transaction specifications. Therefore, the proposed model modifies the shape of data within the blocks, since the regular form of transaction is not required for all data, operations, and processes. Thirdly, it also introduces a modified form of PoS that implies the fairness of the proof-of-queue protocol (PoQ). The proposed protocol is called fair-proof-of-stake (FPoS). Actually, in PoS, the chance of creating and validating blocks is given according to the amount of stake the miner puts in, which causes difficulty for new miners. The PoQ, on the other hand, queues miners and gives a fair chance to every miner. The proposed FPoS uses the block creation and correctness procedure of PoS for a specific number of cycles; then, it gives chances to the miners waiting in the queue (but they cannot compete in the PoS manner). Fourthly, it presents a modified procedure for block verification that relaxes the verification for some trusted transactions and blocks. Obviously, all proposed modifications help to reduce costs, support processing speed-up, and enhance security and privacy. The experimental results show that the total execution time using the modified blockchain was reduced by about 49% comparing to traditional blockchain model. Additionally, the number of messages using modified blockchain was reduced by about 53% compared to the traditional blockchain model.

The rest of this paper is organized as follows: Related work is described in Section 2. Section 3 shows the analysis of healthcare specifications. Section 4 introduces the modified blockchain model that suits e-healthcare specifications. Section 5 illustrates the numerical analysis and the experimental results. Section 6 discusses the advantages of the proposed model, while Section 7 concludes the paper.

2. Related Work

Blockchain technology was used firstly in Bitcoin cryptocurrency [16], and then in other cryptocurrencies such as Ripple, Litecoin, Ethereum, and Zcash [17]. It was introduced to avoid the centralization and control of third parties [18]. Since then, researchers have applied blockchain technology in many other fields, such as healthcare, education, judiciary, etc. [19,20].

Blockchain-distributed architecture is supported by consensus protocols to ensure the correctness of the processes. Different consensus protocols are used, such as PoW, which applies the solutions of some mathematical puzzles with some specifications. The results of such mathematical puzzles are used to hash the proposed block, and the miners use them to verify the correctness of the block [5]. Another consensus protocol is PoS, by which miners use their own coins as guarantees and according to which they have the chance to propose or validate the blocks, which also gives them a chance to win more coins [5]. Proof-of-space allows users to propose their own hard disks and hardware to process and secure the blockchain. According to the given space, the miner has a higher chance. Another protocol is the practical Byzantine fault tolerance (PBFT) consensus protocol, where the number of fault votes should not exceed one-third of the votes [21].

In addition, wearable devices help to sense, collect, and send data and to receive alerts, as well as to share updates and information. This improves healthcare processes and services for all stockholders [11]. Many research has investigated the use of wearable devices in healthcare for patient monitoring [22], recognition, and assistance, as well as for research purposes [23].

Many researchers have linked blockchains with wearable devices to support users, healthcare providers, and insurance companies [24–26]. In such research, the advantages of using blockchain with wearable devices, such as decentralization, distribution, transparency, robustness, availability, automation, traceability, reliability, ownership protection, privacy, and security are investigated, and some of these advantages intersect with each other. In contrast, some work has highlighted the blockchain's disadvantages, such as energy consumption, computational cost, traffic flow latency, and scalability [27]. In response, many works have provided modified blockchain models [28] and modified transactions [29]. Blockchains have been used in healthcare systems for storage security [30], EPHR sharing [31], insurance processes [32], pharmaceutical supply chains [33], patient monitoring [34], organ transplant management [35], clinical trial support [36], and IoT data management [37]. However, to the best of our knowledge, this is the first work that has targeted e-healthcare authority levels, in addition to reducing latency and processing time, using a relaxed blockchain model.

3. Healthcare System's Specifications

Before we move on to integrating blockchain technology with a healthcare system, this paper illustrates some important points related to the healthcare system. Such an analysis allows us to frame the specifications of healthcare data and processes. Accordingly, the blockchain model will be adjusted. The analysis of healthcare specifications is presented as follows:

- *Healthcare system's stockholders*: The main healthcare system's stockholders are patients, doctors, nurses, dentists, health technicians, and administration staff [38].
- *EPHR general privacy*: Patients' information in EPHRs can be revealed for specific purposes, but only if the personal identities are hidden [39]. Such information can be revealed for specific purposes, like research or awareness-raising campaigns. Dealing with EPHRs is very sensitive, even with hidden identities, as they could be negatively exploited by politicians, marketplaces, or businesses.
- *EPHR with healthcare stakeholders*: No patients' information and EPHRs should be hidden form doctors, nurses, dentists, or health technicians [39]. The data can be accessed under a non-disclosure agreement. However, some EPHR information, such as identities or mental and psychological issues, should be hidden from co-members such as volunteers and students who join healthcare teams.
- *EPHR privacy levels*: Different privacy levels are assigned to EPHRs. For example, EPHRs would require specific privacy levels [39,40] for politicians, military leaders, and famous people compared to the public.
- *Updating EPHR*: Different parts of EPHRs can be updated by doctors, nurses, dentists, or health technicians. Indeed, everyone who is part of the healthcare staff can update specific related parts without restriction. However, we should restrict updates that are irrelevant to the roles within the healthcare team [41].
- *Direct update of EPHR*: Most EPHR updates require the approval of the primary or main doctor. Primary doctors do not need any approval, and they can authorize others to directly update the relevant parts or sections without any approval [42].
- *Financial Transactions*: Financial transactions can be accessed or updated by the authorized people, although the approval of any operation is required [32].
- *Latency*: Latency or delays that result from approval are critical in some emergency cases and scenarios.
- *System considerations*: The development of any healthcare system should consider the general regulations, ethical obligations, and cultural influence. For example, cases of abortion, transgender status, and violence should be treated according to the laws and culture perspectives, which differ from one place to another.

4. Modified Blockchain Model

Blockchain technology introduces decentralized and distributed architecture that is combined with supported algorithms and procedures. The blockchain has nodes that are fully connected to each other and share copies of the same ledger. The algorithm starts by confirming the correctness of transactions and groups them in blocks. Only one of the miners can propose a new block. This miner is decided based on different consensus mechanisms, such as PoW, PoS, PoQ, or PBFT. Then, the other miners validate and vote on the approval of the proposed block. According to the votes of the majority (consensus), the block is approved and added to ledger; otherwise, it is ignored and another block is proposed. In fact, every miner has to update their copy the ledger based on the consensus.

In this paper, the blockchain procedure is modified according to the guidance of the results of the healthcare specifications in Section 3. The details of the modifications are explained in the following subsections.

4.1. Transaction Process

Understanding the nature of transactions is important. Unfortunately, many works apply blockchain technology without studying the implication of using transactions, which obviously indicates a lack of understanding of transaction specifications.

A regular operation reads data or updates it directly in the main memory. However, a transaction consists of one or more operations. These operations involve either reading a piece of data or writing a piece of data. The operations within a transaction are executed one by one in a temporary memory (buffer). In the end, the transaction is committed or aborted. Committing a transaction means that the results of all operations are reflected from the temporary memory to the main memory, while aborting a transaction means that the results of all operations are neglected and the temporary memory is freed. In addition, by using a transactional system, many operations and transaction are executed in parallel, which increases the throughput (number of executed transactions per time). It also speeds up the processing system with minimal costs. Another advantage is the ability to roll back and retrieve the correct data. The transactional system is supported by software and hardware resources.

4.2. Mining

The mining process has two basic steps. Firstly, the correctness of the transactions is confirmed, and they are placed into a new block. Secondly, a consensus protocol such as PoW or PoS is followed to obtain a chance of proposing the new block to others for validation and votes, which is explained in the following section. Now, let us focus on the correctness of transactions; indeed, there are two kinds of transactions: read and update. The read transaction only includes read operations, while the update transaction includes at least one write operation, as shown in Figure 1. Actually, Figure 1 shows examples of different kinds of transactions. T1 is a read transaction that includes only one read operation, which returns the value of the variable *a*. At the end, the transaction tries to commit (TryC). T2 is a read transaction that includes multiple read operations for the variables *a*, *b*, and *c*. T3 is an update transaction that includes only one write operation, which updates the value of the variable *a* with the value 5. T4 is a update transaction that includes multiple write operations to update the variables *a*, *b*, and *c*, consequently with the values 1, 2, and 3. T5 is an update transaction that includes multiple read and write operations, which read variable *a*, write to *a*, then read variable *c*.

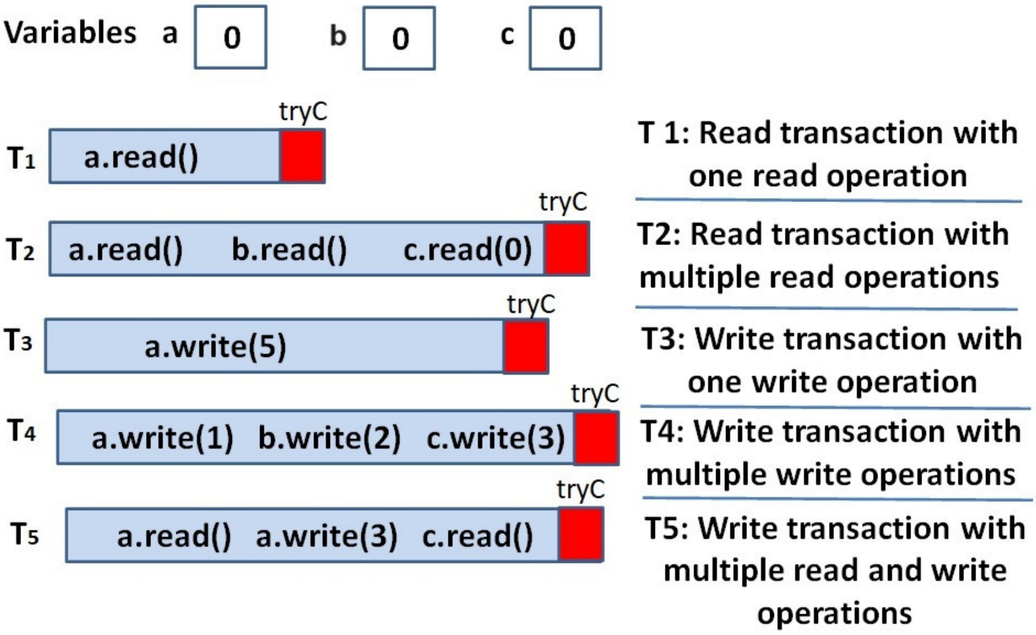
Variables a 0 b 0 c 0
T1
a.read()
tryC
T 1: Read transaction with one read operation
T2
a.read() b.read() c.read(0)
tryC
T2: Read transaction with multiple read operations
T3
a.write(5)
tryC
T3: Write transaction with one write operation
T4
a.write(1) b.write(2) c.write(3)
tryC
T4: Write transaction with multiple write operations
T5
a.read() a.write(3) c.read()
tryC
T5: Write transaction with multiple read and write operations

Figure 1. Kinds of transactions.

Furthermore, since transactions run in parallel and in isolation, the effects of the operations appear after transaction is committed. This means that the changes that result from transaction operations are not visible to the system until it commits. This may allow for inaccurate data to be read (not up to data), and may cause conflict among transactions, as illustrated in Figure 2. In fact, running a read transaction in parallel does not have any negative impact, since the values of the memory data are stable. However, update transactions change the value of the memory content. Figure 2 gives an example of four transactions running in parallel and shows how the conflicts caused the abortion of some transactions. T1, T2, and T3 are read transactions, while T4 is an update transaction. The operations of those transactions accessed some memory variables, namely, a = 0, *b* = 0, *c* = 0, and *d* = 0. Actually, T1 read variable *a* (returning *a* = 0), *c* (returning *c* = 0), and *d* (returning *d* = 0). T2 read variables *a* (returning *a* = 0) and *b* (returning *b* = 1), which means that T4 changed the value of *b* and had already committed. T3 read variable *b* (returning *b* = 0) and *d* (returning *d* = 1). This means that T3 read *b* before the commitment of T4, and *d* afterward. T4 read variable *a* (returning *a* = 0) and updated the values of *b* (returning *b* = 1) and *d* (returning *d* = 1).

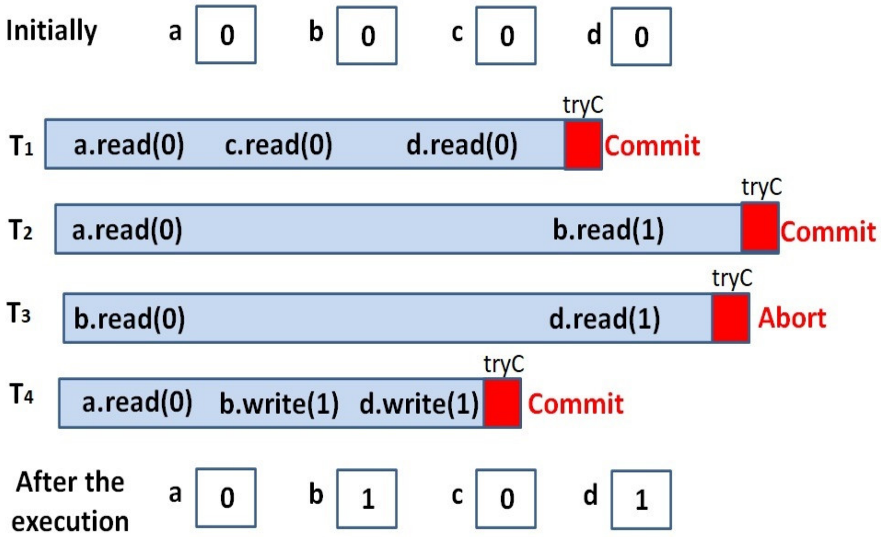

Figure 2. An Example of parallel execution of four transactions to illustrate the conflicts that caused the abortion of T4.

In fact, the correctness of the execution of parallel transaction is confirmed when the results of parallel execution match any correct serialized execution. In other words, the transactions in parallel execution must be ordered in a logical way. Considering our example in Figure 2, T1 can be considered as the first executed transaction, since it returned all original values of *a*, *c*, and *d*. T4 can be considered as the second executed transaction, since it read the original values of *a* and updated the value of *b* (*b* = 1) and *d* (*d* = 1). T2 can be considered as the third executed transaction, since it read the original values of *a* (no concurrent transaction has updated *a*) and *b* (returning *b* = 1), where the *b* value had been updated by T4. However, T3 must be aborted and ignored in the process of ordering; it could not be ordered before T4 as it read *d* = 1, which could only be seen after the commitment of T4, and it could not be ordered after T4 as it read *b* = 0, clearly ignoring the change in variable *b* that took place after T4 was committed.

One last example is shown in Figure 3 to explain the conflicts between two update transactions. Let us have one bank account, called the hospital bank account (*h*), and let *h* = 10,000, which means that the hospital bank account contains USD 10,000. Now, let there be two transactions, T1 and T2, that belong to different patients and are executed in parallel. The first patient pays USD 100 as the cost of some blood tests (T1), while the other pays USD 500 for medical examination (T2). In this way, T1 should read the value of *h* and add a value of 100, while T2 should read the value of *h* and add a value of 500. Since both T1 and T2 are executed in parallel, both of them will read the original value of *h*. This means that T1 will add 100 to 10,000, and at the end, it will write *h* = 10,100, while T2 will add 500 to 10,000, and at the end it writes *h* = 10,500. The last results of both transactions are incorrect, since the value of *h* after the commitment of T1 and T2 should be 10,600. Consequently, one of the transactions must be aborted and rolled back, then executed again after the commitment of the other. Let us say that T2 is aborted, then executed again after the commitment of T1: it sees *h* = 10,100, and by adding 500, the result becomes *h* = 10,600.

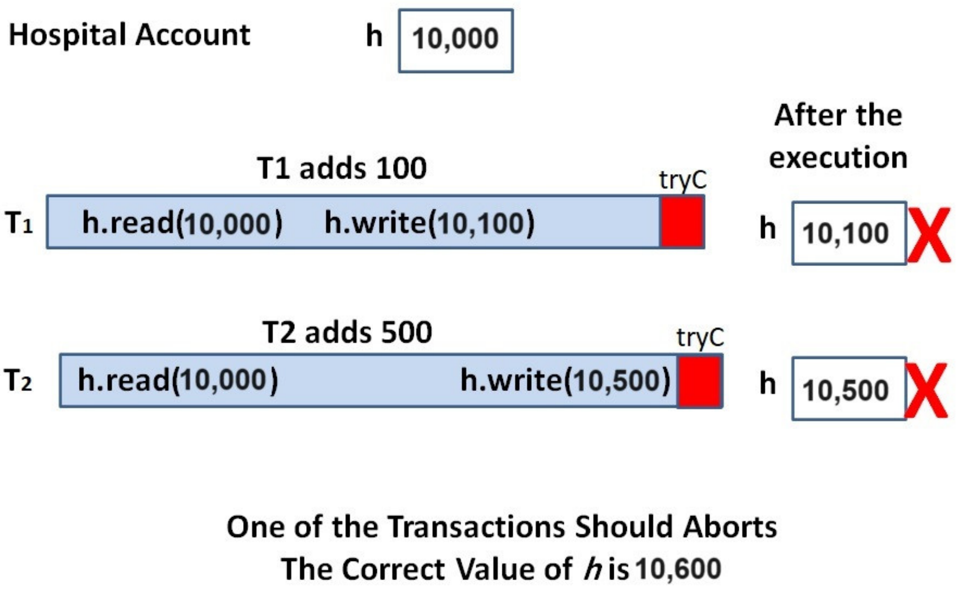

Figure 3. The conflict between two update transactions.

Aborting some transactions to guarantee correctness is one of the disadvantages of transactional processing systems, because it costs time, energy, effort to maintain and it does not suit healthcare systems. To cope with such an issue, some works relax the correctness property as they allow us to read some stale data (not up to date) [43], but such strategies might be harmful in cases of sensitive data, such as in the healthcare system.

4.3. Fair-Proof-of-Stake

In addition, we introduce the concept of fair-proof-of-stake (FPoS). Actually, the traditional PoS is a consensus mechanism that allows a miner to create a new block and validate the proposed block according to the number of coins a miner stakes. A miner who stakes a large amount has more to lose it in case the block has not been verified. This enhances the confidentiality of miners and the security of the system. However, new miners who have just joined the blockchain usually do not have enough to stake or to compete with old miners. On the other hand, PoQ queues miners in a fair manner, for example, using timestamps. However, all miners have the same chance regardless of their trustworthiness or confidentiality. For such issues, FPoS is proposed to balance the ideas of PoS and PoQ. The system assigns a ratio—say, x:y—such that the system applies PoS for x cycles, then applies PoQ for y cycles. For example, if the ratio is 10:1, then the one with higher stakes takes a chance, and this is repeated 10 times. Then, at cycle number 11, the first miner in the queue (it might be a newly joined miner or a miner with few stakes) is given a chance to create or validate a block. Obviously, the ratio can be changed over time according to the need of the system. In short, FPoS helps to motivate nodes with low processing capabilities or low credit to participate in mining processes; gives more of a chance to the trustworthy miners; and provides flexibility, as the ratio is decided according to the system's nature and data sensitivity.

Algorithm 1 shows FPoS, setting the FPoS ratio x:y (Line 2). At the beginning, if any node N_i wants to be selected as a miner, it has to stake some coins to join the stake list (*flag* = 1), or it joins the queue (*flag* = 2). Any other value of the *flag* will lead to an error (Lines 7–15). Then, FPoS selects a miner from the stake list x times. Every time, it selects the node of the maximum stake (Lines 17–19). After that, y times, it selects a miner from the queue (Lines 20–22). Finally, it starts over and sets the counters to zero (Lines 23–24).

Algorithm 1: FPoS

```
1.  // Initialization:
2.  Input: x, y; // Insert x:y ratio for FPoS
3.  counter1 = 0;
4.  counter2 = 0;
5.  flag; // Flage to join miner lists
6.  // Node $N_i$ joins miner lists by inserting 1 to stake coins or 2 to joins Queue;
7.  Switch (flag):
8.  Case flag = 1;
9.          $N_i \rightarrow$ Stake(); // Node stakes some coins
10. break;
11. Case flag = 2;
12.         $N_i \rightarrow$ Queue; // Node is enqueued
13. break;
14. default:
15.       error(); // flag $\neq$ 1 or 2
16. // Now the FPoS selects miners
17. While counter1 < x;
18.         miner$\leftarrow$maxStake();
19.         counter1++;
20. While counter2 < x;
21.         miner$\leftarrow$Queue();
22.         counter2++;
23. goto(line 3);
```

4.4. Trusted Transactions

In healthcare systems, there are different kinds of operations and privileges. Firstly, some members of medical team are authorized to read and update specific parts of EPHR. Therefore, they can access their parts directly, without permission (before the access) or approval (after the execution of operations). This means that they can execute and update memory directly in a non-transactional manner. Our model keeps them in a transactional manner, but it relaxes the validation process for such transactions. This is called a level-one trusted transaction, denoted as *TrustedT1*. Secondly, some healthcare practitioners can access data for reading, but any updates require approval by a primary doctor, for example. This form can be called a level-two trusted transaction, denoted as *TrustedT2*. The third kind of process requires approval for both reading and updating, which can be executed in a regular transaction manner denoted as *T*. In fact, the data that come from IoT or wearable devices is processed using regular *T*. Moreover, coordinating parallel transactions can be solved through some efficient leader election algorithms [44], where the primary doctor acts as the leader of the medical team and organizes the access to the EPHR.

Another important issue is focusing on the content of the transaction. One example is the transaction of Bitcoin, which includes information about the sender, receiver, and amount. There are different patterns of transactions in e-health systems, such as read transactions and update transactions for EPHR, as well as financial transactions for insurance and other related process. Unlike Bitcoin's transactions, where the sender and receiver identities appear to the public, in healthcare transactions, patients' identities should be hidden for privacy purposes using smart contracts and encryption, as explained later on in Section 6.2.

4.5. Blocks and Trusted Blocks

Having different kinds of transactions allows us to have different kinds of blocks. There are three kinds of blocks: *Block1* includes *TrustedT1*, *Block2* includes *TrustedT2*, and regular *Block* includes *T*. Since *TrustedT1* transactions do not need approval, miners do not need to spend time or effort to verify them. They just checks the authorization of the performer of the transactions, and so they receive the minimum reward (which can be some amount of coins of any cryptocurrency). Moreover, *Block2* includes *TrustedT2*. For *TrustedT2*, read transactions do not need approval and are treated as *TrustedT1*. This only requires verification that the transaction has already been approved by the primary doctor, so miners receive an average amount in reward. Update transactions in *TrustedT2,* however, require the correctness of the transactions to be validated and the approval to be verified, and the miner receives a higher reward. However, miners receive the maximum reward for verifying *T* and, consequently, *Block*. Figure 4 shows a copy of a ledger that has a chain composed of different kinds of blocks. It also shows that each type of block has the hash of the previous block, the hash of the current block, and specific kinds of transactions. Indeed, *Block1* includes *TrustedT1*, *Block2* includes *TrustedT2*, and regular *Block* includes regular transactions (*T*).

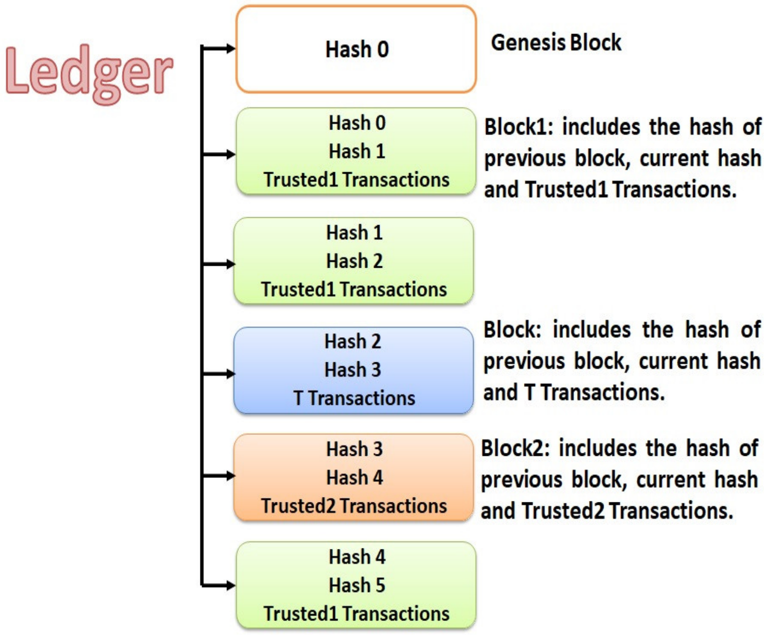
Ledger
Hash 0
Genesis Block
Hash 0
Hash 1
Trusted1 Transactions
Block1: includes the hash of previous block, current hash and Trusted1 Transactions.
Hash 1
Hash 2
Trusted1 Transactions
Hash 2
Hash 3
T Transactions
Block: includes the hash of previous block, current hash and T Transactions.
Hash 3
Hash 4
Trusted2 Transactions
Block2: includes the hash of previous block, current hash and Trusted2 Transactions.
Hash 4
Hash 5
Trusted1 Transactions

Figure 4. The ledger in the proposed blockchain model, containing three kinds of blocks, namely, Block, Block1 and Block2. Every kind of block includes a specific kind of transaction.

4.6. The Proposed Algorithm

The blockchain algorithm consists of three stages: proposing, voting, and consensus. The proposing stage includes transactional correctness, block creations, hash calculation, and casting a new block through FPoS to the others. All proposing steps have already been explained in the previous sections. The current section shows the validation and consensus processes, as well as the preliminaries and the algorithm.

4.6.1. Proposal Sending and Validating

At this level, the validator group is prepared. The validator group includes the number of miners who are going to validate the proposed block. The process includes sending the proposed block to every member of the validator group. Clearly, having a large number of validators enhances the accuracy and correctness of the validation process, but it also uses a large number of communication messages and increased processing. In our model, *Block1* requires the minimum number of validators—say, three validators. The odd number of validators helps to satisfy consensuses. *Block2* requires an average number of validators, while a regular *Block* require more validators. As mentioned earlier, the validators are selected according to the FPoS procedure. After nominating the members of the validator group, the proposed block is sent to all members.

Upon receiving the proposed block, every member validates the block and casts the result to all nodes. The result is either 1, for a valid block, or 0, for an invalid one. Actually, in a regular validation process, the validator checks the proposer's identity (to confirm the authorization), as well as whether the block has been modified after the proposing step, which can be assessed through the block's hash validation. However, validating the content of the block (transactions) is an important step to cope with the sensitivity of healthcare data. In fact, to validate *Block1*, the validators do not validate the transactions, but instead just check that all transactions have been executed by primary doctors. To validate *Block2*, the validator should check that all read transactions have been executed by authorized members and that all update transactions are correct, updated by authorized members, and approved by primary doctors. For regular *Block*, the validator should check that all transactions are correct, executed by authorized members, and approved by primary doctors. Finally, every validator casts the validation results (vote).

4.6.2. Consensus

Upon receiving the validation results from all validators, every node counts the valid and invalid votes to calculate the votes of the majority. According to the majority, if the block is valid, it is linked to the ledger; otherwise, the block is rejected and ignored. Indeed, the consensus is a procedure that can vary from one system to another. For example, a consensus of the majority can be reached by confirming a mathematical puzzle or by reaching a majority of 51% of the votes, while in other systems, it requires at least 2/3 of votes, which suits our model. The sensitivity of healthcare data and the nature of technology require higher percent for the consensus procedure.

4.6.3. Preliminaries and Notations

As shown in Algorithm 2, the blockchain process starts with node N_i, which mines T_i transactions to validate them, and then fills a new block B_i with the validated transactions considering the block size *Bsize*. Indeed, there are three kinds of transactions and, consequently, three kinds of blocks (Lines 9–29). Then, nodes compete to find hash and propose a new block using FPoS (Lines 31–33). Next, the blockchain algorithm assigns the validator group *voters* according to the kind of block using *validateor_size()*, then casts the block (Lines 34–35).

As mentioned earlier, according to the block type, validators have different levels of validation processes (Lines 37–44). They cast their votes (Line 45), which are either 1 for correct or *0* for not correct, as shown in Algorithm 3.

In Algorithm 4, upon receiving the votes, the consensus decision is made according to the votes of the majority to commit the new block and either add it to L_i or abort it.

Algorithm 2: Modified Blockchain Algorithm (Proposing)

```
1.   // Initialization:
2.   L; // The current ledge
3.   T; // Transaction
4.   B; // The current block
5.   Bsize← $x$; // $x$ is an integer representing the block size

6.   Proposing stage:
7.   // Create $B_i$
8.   // For any node $N_i$: decides what kinds of transactions to validate;
9.   Switch ($T_i$):
10.  Case TrustedT1:
11.      For j = 0 to j < Bsize
12.      if (valid TrustedT1 == True)
13.          TrustedT1→Block1;
14.          $B_i$→Block1;
```

Algorithm 2 *Cont.*

```
break;
Case TrustedT2:
    For j = 0 to j < Bsize
    if (valid TrustedT2 == True)
        TrustedT2→Block2;
        $B_i$→Block2;
break;
Case T:
    For j = 0 to j < Bsize
    if (valid T == True)
        T→Block;
        $B_i$→Block;
break;
default:
    exit(0);

// After $B_i$ is created, hash $B_i$, use FPoS, decide validators group and cast $B_i$
$B_i$→hash();
While (($B_i$→FPoS()) == false)
        Wait();
validateor_size($B_i$);
Cast($B_i$);
```

Algorithm 3: Modified Blockchain Algorithm (Voting)

```
// According $B_i$ Kind determines the level of validation, then vote
Case $B_i$ →TrustedT1:
    Low_validation();
break;
Case $B_i$ →TrustedT2:
    Normal_validation();
break;
default:    // When $B_i$ →T
    high_validation();
Vote();
```

Algorithm 4: Modified Blockchain Algorithm (Consensus)

```
// According to the vote of majority $B_i$ is added to $L_i$ or is aborted
if (majority() == True)
    message(Commit);
    $B_i$ →$L_i$
else
    message(Abort);
```

5. Experimental Results

In this paper, the blockchain procedures are modified according to the guidance of the results of the healthcare specifications. The experiment focuses on a modified blockchain for healthcare, which relaxes blockchain procedures for some blocks. As mentioned earlier, the modified blockchain has three kinds of blocks depending on what kind of transactions are in the block. The experiment simulates traditional blockchain procedures and modified blockchain procedures. It compared the performances of both, focusing on total execution time. The experiment was run on OS-Window 10 using Java as the programming language and an Intel Core (TM) i7 CPU, with 2.90 GHz and 4 GB (RAM). Every test was run for five times, and the average is shown.

5.1. Execution Time

In fact, the experiment created 12,000 transactions, including 6000 read transactions and 6000 update transactions. The read transactions were divided into three categories: *TrustedT1* (2000 transactions), *TrustedT2* (2000 transactions), and *T* (2000 transactions); the same categories are applicable to the update transactions. After the execution of the transactions, the blockchain procedure was executed to verify the transactions and store the verified ones in a blockchain ledger.

For simplicity, in this experiment, we built a blockchain of ten nodes. The block creation was fairly distributed among the ten nodes, as every cycle-one node proposed a new block of size 50 transactions (every block contained 50 transactions). The block also had a hash number which represented the content of the block, including the identity of the creator node, for authentication. After the block creation step, it was sent to the other nodes for verification.

Actually, a traditional blockchain treats all transactions with the same procedures, while a modified blockchain relaxes the validation of all *TrustedT1* and the read transactions from *TrustedT2*. The update transactions from *TrustedT2* and all *T* were treated as in a traditional blockchain. Consequently, the execution time of *Block1* was the minimum, the execution time of *Block2* was average, and the execution time of *Block* was the maximum. Figure 5 shows the total execution time for the traditional blockchain and modified blockchain, where the modified blockchain reduced the execution time by about 49% since about 33% of blocks were the *Block1* type, about 33% of blocks were the *Block2* type, and about 33% of blocks were the *Block* type, while all blocks of the traditional blockchain were the *Block* type.

In addition, Figure 6 illustrates the detailed execution time for the validation process of the read and update transactions for *Block1*, *Block2*, and *block*. It shows how relaxation reduced the execution time. Actually, for *Block1*, it only checked the hash for the block, since the transactions came from trusted executers. The same thing applied to read transactions in *Block2*, but for update transactions, it validated each transaction, then checked the hash for the block. For *block*, it validated each transaction (both the read and update ones) and the hash for the block.

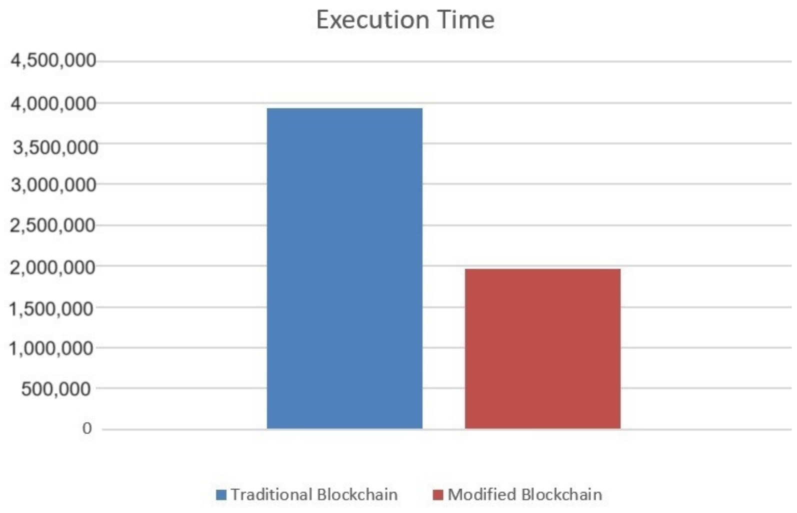

Figure 5. Total execution time for the traditional blockchain and modified blockchain.

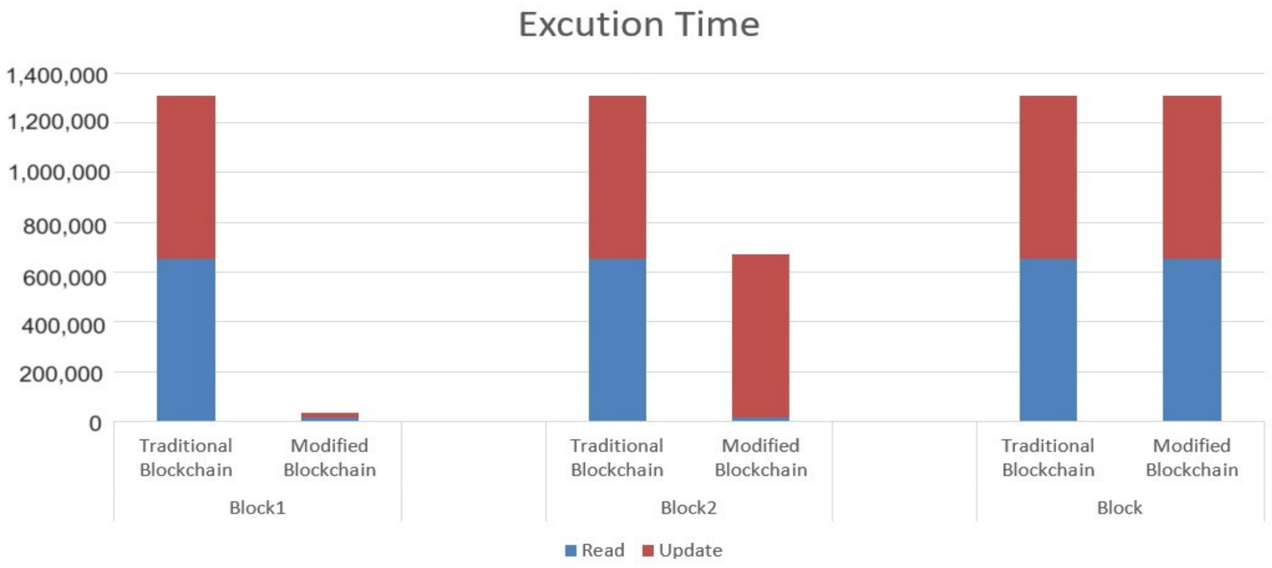

Figure 6. The detailed execution time for the validation process of read and write transactions within Block1, Block2, and Block using a traditional blockchain and a modified blockchain.

5.2. Numrical Analysis for the Number of Messages

The decentralization characteristic of a blockchain requires a large number of messages. In fact, the miner node sends the proposed block to all other nodes for validation; say n nodes require n messages. Then, every node validates the proposed block and sends the vote to all others (excluding itself), which costs $n(n-1)$ messages. After that, every node calculates the majority of votes and send the consensus decision to all others (excluding itself), which costs another $n(n-1)$ messages. The number of messages for this blockchain process is presented below:

$$b(n + (2(n(n-1)))) \tag{1}$$

where b is the total number of blocks and n is the number of nodes.

The modified blockchain reduces the validation process as well as the number of validators for *Block1* to the minimum—say, only three nodes (in our proposal).

It also reduces the number of validators for *Block2* to the average—say only six nodes (in our proposal)—while the number of validators for *Block* will be ten for both the traditional and the modified blockchain. Using our experiment's specifications, there were 4000 *TrustedT1* (both reads and updates), 4000 *TrustedT2* (both reads and updates) and 4000 *T* (both reads and updates). Since the block size was 50 transactions, there were 80 *Block1*, 80 *Block2*, and 80 *Block*. Figure 7a shows the relationship between the number of messages and the number of nodes where the number of blocks was constant (50 blocks). It highlights the total number of messages for *Block1*, which was 1200 messages; 5280 messages for *block2;* and 15,200 messages for *Block.* By increasing the number of nodes, the number of messages increased exponentially. Figure 7b demonstrates the relationship between the number of messages and number of blocks where the number of nodes was constant (10 nodes). By increasing the number of blocks, the number of messages increased linearly.

Figure 8, illustrates the total number of messages the modified blockchain which includes 80 *Block1*, 80 *Block2* and 80 *Block*, 240 total. It also shows the total number of messages for the traditional blockchain that includes 240 blocks of type *Block*. Obviously, the total number of messages using modified blockchain is reduced by about 53% compared to the traditional blockchain model.

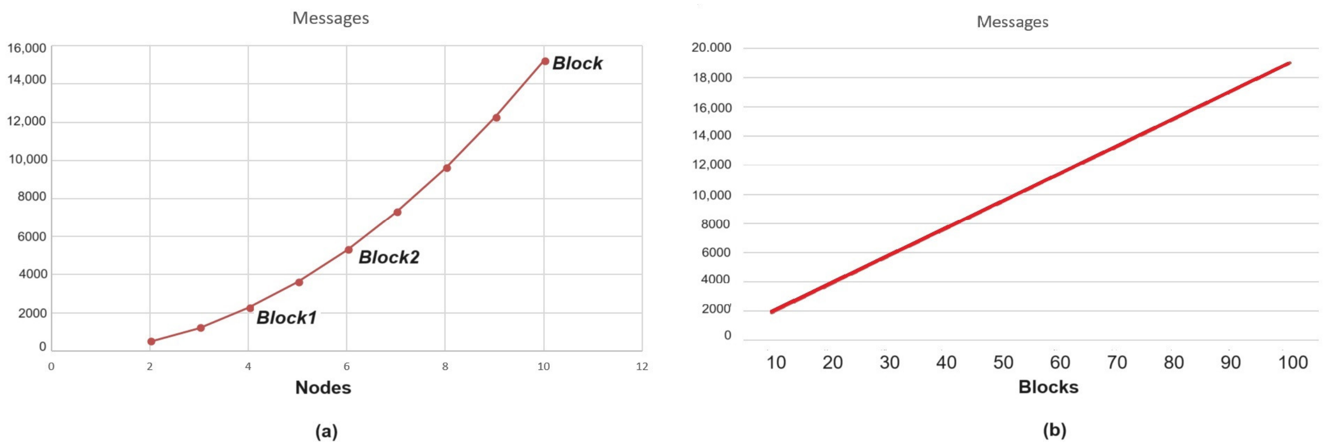

Figure 7. (**a**) Shows the Relationship between Number of Messages and Number of Nodes where the Number of Blocks is Constant, Highlighting the Number of Messages for Block1, Block2 and Block; (**b**) Shows the Relationship between Number of Messages and Number of Blocks where the Number of Nodes is Constant.

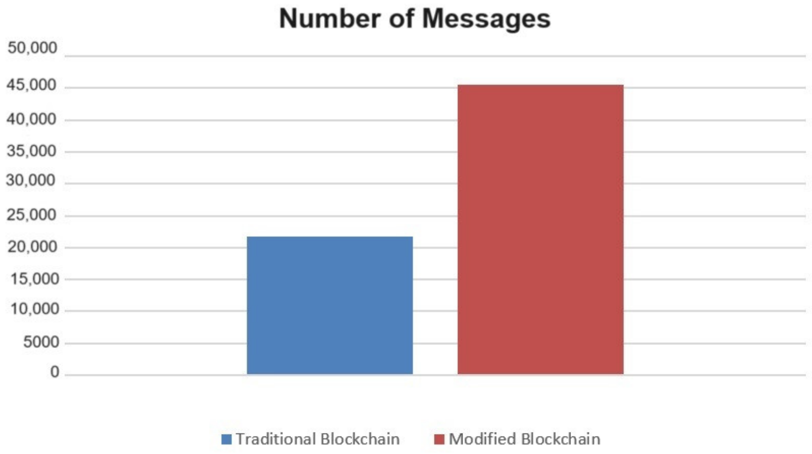

Figure 8. The number of messages using traditional blockchain and modified blockchain.

6. Discussion

This section discusses blockchain technology in general, as well as the proposed modified blockchain that is illustrated in this work. It also discusses the IoT and wearable device specifications which may enhance our model.

6.1. Decentralization

One of the main characteristics of using blockchain technology is decentralization, where the blockchain eliminates the central authority, since every node and device in the blockchain network can generate data and all decisions are made according to the consensus. This allows for fault tolerance, as it does not have a single point of failure. This means the system keeps running even if there are some faulty or failed components. In addition, every node in the blockchain has an up-to-date copy of the data, which helps to avoid centralized data storage. Actually, this supports data availability, accessibility, and recovery if needed [20]. However, it costs more space, time, computation, communication, and energy to manage the data's consistency.

6.2. Security

Another important feature of blockchain technology is security, which also includes authentication, authorization, and privacy. The structure of a blockchain allows us to have multiple copies of data ledgers, which facilitates discovery and recovery in case of data misusing or corruption issues. Every block in the ledger is secured through a hash number that considers the content of the current block and the hash of the previous block in the ledger. This prevents any change in the ledger or the blocks as the chain of the hash numbers is changed, which can be directly discovered, and the approval of such a change is reached through consensus, which makes it impossible.

Moreover, a blockchain allows direct communication between any two nodes, which helps to authenticate every communication. The authentication is performed using smart contracts and encryption. Smart contracts include some conditions, and if those conditions are satisfied, specific actions are taken. It is an automatic code that is immutable (cannot be changed) and distributed (so every node knows the actions it takes) [26]. In our model, every node and device in the blockchain has a unique ID, authority, privilege, and role, which are listed in a smart contract. Upon any communication, the nodes or devices authenticate each other through smart contract. This also helps to validate whether the communication or the requested action is authorized. This is implemented using a private or public key, which allows for validated authorization of access or updates to any data. This guarantees privacy and control. Furthermore, the transferred data and patient identities are encrypted for privacy purposes. In fact, according to the healthcare specifications, there are different levels of data sensitivity, and there are specific considerations for some patients, such as some governmental or military officials. Thus, different levels of security protocols, encryption techniques, and privacy procedures can be applied.

6.3. Performance

It is important to discuss the performance in cases of applying blockchains in healthcare systems. As mentioned earlier, applying blockchain technology has a negative impact on the space, as a copy of the data is attached to every node. It also costs a large number of messages exchanges to perform blockchain tasks such as proposing, voting, and consensus. In addition, the cost of computation increases as a result of decentralization, in which every validator runs validation processes and processes for consensus calculations [1].

On the other hand, the use of blockchain technology results in many advantages that enhance its performance. In fact, having direct communication among nodes and devices speeds up communication. In addition, having smart contracts within a blockchain enables the automation of processes and actions. In fact, smart contracts and consensus decisions both give more reliability to blockchain technology. Another vital advantage of blockchain technology is scalability, where large numbers of IoT sensors, wearable devices, users' devices, and others can be adapted. Blockchains also facilitate the management, coordination, and cooperation of heterogeneous components within systems. One of the main advantages is the ability to manage the components and identities of users, processes, services and ownerships. Those elements interweave many processes and are subject to change over time, which is a challenging task.

Moreover, the procedures of validation in blockchains clearly challenge attempts at manipulation. Such features reduce the chances of aborting transactions and blocks, which improves the performance, since these abortions waste execution time. Indeed, all aborted transactions should be rolled back and re-executed again. Some studies have shown that the abortion of read transactions costs about 80% of the total execution time [43].

6.4. Sensors and Wearable Devices

The development of sensors and wearable devices has enhanced their uses in many fields of technology. There are various categories of wearable devices, i.e., clothes like smart shirts, accessories such as watches and earbuds, and embedded wearable devices such as biosensors and smart tattoos. The use of wearable devices has many advantages, such as providing real-time data and responses. It supports data accessibility, availability, automation, and control. On the other hand, wearable devices cause some weaknesses to the systems from the point of view of security, energy, and computation abilities. Actually, most wearable devices use Bluetooth technology for communication, which is considered a vulnerable communication technology from a security perspective. Therefore, it should be supported with advanced security protocols and techniques for key generation, key characteristics, key exchange, and data encryption, as well as to detect malicious and infected devices.

However, having advanced security protocols and techniques is challenged by computation and energy capabilities that are limited as a result of the sizes of wearable devices [15]. In this case, the system should balance the computation and energy capabilities on one side, and security on another side. It also should balance the computation and energy capabilities with the quality of data. Actually, to overcome the challenge of computation, it is recommended to exploit the computation power of other system components or faraway servers [45,46]. It could also outsource the computation and processing tasks in case the data are not sensitive. Moreover, for energy consumption, there are different kinds of energy technology that can be involved in enhancing wearable devices, such as renewable energy that comes from nature (e.g., sun, wind, and body temperature).

6.5. Risk Management

At the end this discussion, it is important to highlight risk management for the modified blockchain model. Firstly, blockchain technology promises to change centralized management by enforcing democratic decision-making techniques through data sharing and voting. However, the modified blockchain model satisfies healthcare specifications by reducing data sharing and voting in response to the authority level and case sensitivity. Such a reduction risks the reliability of the decision-making. Secondly, people's identities are another critical issue, as hiding the identities of doctors would hide their authority levels, which would risk the relaxation of the validation process. Thirdly, using security techniques such as hashing and encryption risks delays data accessibility, which is critical in some healthcare cases. Fourthly, the cultural resistance to the new methods of processing and management that restrict privileges is another challenge. Finally, the automation of data access, processing, communication, and voting is very difficult to adopt in some areas of healthcare system because it may risk lives.

7. Conclusions

Blockchains are a promising type of technology to be applied in healthcare systems. This work introduces a modified model of a blockchain to enhance e-healthcare with blockchain advantages. This work investigates healthcare and transaction specifications to modify the blockchain model. Indeed, it proposes different kinds of transactions and blocks which enable different levels of processing, validation, and security. This allows us to improve the performance, to increase efficiency, and to reduce costs. For future work, it is still challenging to compromise blockchain reliability to reduce time and communication costs. Moreover, it is important to develop a blockchain model that reduces space complexity and energy consumption. It also should enhance wearable device security and computation abilities. These fields will need to discover specific techniques for such improvements.

Funding: This research received no external funding.

Acknowledgments: We would like to acknowledge the assistance of Medical College members at Jazan University for the fruitful sessions of discussions that helped us in understanding and analyzing the roles, authority, data, and processing of healthcare systems.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Assiri, B. October. Leader Election and Blockchain Algorithm in Cloud Environment for E-Health. In Proceedings of the 2019 2nd International Conference on new Trends in Computing Sciences (ICTCS), Amman, Jordan, 9–11 October 2019; pp. 1–6.
2. Kuo, M.H. Opportunities and challenges of cloud computing to improve health care services. *J. Med. Internet Res.* **2011**, *13*, e1867. [CrossRef] [PubMed]

3. Dinh, H.T.; Lee, C.; Niyato, D.; Wang, P. A survey of mobile cloud computing: Architecture, applications, and approaches. *Wirel. Commun. Mob. Comput.* **2013**, *13*, 1587–1611. [CrossRef]
4. Ghosh, P.K.; Chakraborty, A.; Hasan, M.; Rashid, K.; Siddique, A.H. Blockchain application in healthcare systems: A review. *Systems* **2023**, *11*, 38. [CrossRef]
5. Xu, J.; Wang, C.; Jia, X. A survey of blockchain consensus protocols. *ACM Comput. Surv.* **2023**, *55*, 278. [CrossRef]
6. Guo, H.; Yu, X. A survey on blockchain technology and its security. *Blockchain Res. Appl.* **2022**, *3*, 100067. [CrossRef]
7. Assiri, B.; Khan, W.Z. Fair and trustworthy: Lock-free enhanced tendermint blockchain algorithm. *TELKOMNIKA (Telecommun. Comput. Electron. Control)* **2020**, *18*, 2224–2234. [CrossRef]
8. Subhan, F.; Mirza, A.; Su'ud, M.B.M.; Alam, M.M.; Nisar, S.; Habib, U.; Iqbal, M.Z. AI-enabled wearable medical internet of things in healthcare system: A survey. *Appl. Sci.* **2023**, *13*, 1394. [CrossRef]
9. Lee, Y.H.; Medioni, G. RGB-D camera based wearable navigation system for the visually impaired. *Comput. Vis. Image Underst.* **2016**, *149*, 3–20. [CrossRef]
10. Jawale, A.S.; Park, J.S. A security analysis on apple pay. In Proceedings of the 2016 European Intelligence and Security Informatics Conference (EISIC), Uppsala, Sweden, 17–19 August 2016; pp. 160–163.
11. Borowski-Beszta, M.; Polasik, M. Wearable devices: New quality in sports and finance. *J. Phys. Educ. Sport* **2017**, *20*, 1077–1084.
12. Vidal, M.; Turner, J.; Bulling, A.; Gellersen, H. Wearable eye tracking for mental health monitoring. *Comput. Commun.* **2012**, *35*, 1306–1311. [CrossRef]
13. Wijsman, J.; Grundlehner, B.; Liu, H.; Hermens, H.; Penders, J. Towards mental stress detection using wearable physiological sensors. In Proceedings of the 2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Boston, MA, USA, 30 August–3 September 2011; pp. 1798–1801.
14. Tyagi, A.K.; Dananjayan, S.; Agarwal, D.; Thariq Ahmed, H.F. Blockchain—Internet of Things Applications: Opportunities and Challenges for Industry 4.0 and Society 5.0. *Sensors* **2023**, *23*, 947. [CrossRef] [PubMed]
15. Yugank, H.K.; Sharma, R.; Gupta, S.H. An approach to analyse energy consumption of an IoT system. *Int. J. Inf. Technol.* **2022**, *14*, 2549–2558. [CrossRef]
16. Nakamoto, S. *Bitcoin: A Peer-to-Peer Electronic Cash System*. Decentralized Business Review. 2008. p. 21260. Available online: https://bitcoin.org/bitcoin.pdf (accessed on 1 October 2023).
17. Andrianto, Y.; Diputra, Y. The effect of cryptocurrency on investment portfolio effectiveness. *J. Financ. Account.* **2017**, *5*, 229–238. [CrossRef]
18. Akcora, C.G.; Gel, Y.R.; Kantarcioglu, M. Blockchain networks: Data structures of Bitcoin, Monero, Zcash, Ethereum, Ripple, and Iota. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2022**, *12*, e1436. [CrossRef] [PubMed]
19. Crosby, M.; Pattanayak, P.; Verma, S.; Kalyanaraman, V. Blockchain technology: Beyond bitcoin. *Appl. Innov.* **2016**, *2*, 71.
20. Kuo, T.T.; Kim, H.E.; Ohno-Machado, L. Blockchain distributed ledger technologies for biomedical and health care applications. *J. Am. Med. Inform. Assoc.* **2017**, *24*, 1211–1220. [CrossRef] [PubMed]
21. Chen, Y.; Li, M.; Zhu, X.; Fang, K.; Ren, Q.; Guo, T.; Chen, X.; Li, C.; Zou, Z.; Deng, Y. An improved algorithm for practical byzantine fault tolerance to large-scale consortium chain. *Inf. Process. Manag.* **2022**, *59*, 102884. [CrossRef]
22. Pantelopoulos, A.; Bourbakis, N.G. A survey on wearable sensor-based systems for health monitoring and prognosis. *IEEE Trans. Syst. Man Cybern. Part C* **2009**, *40*, 1–12. [CrossRef]
23. Baig, M.M.; Gholamhosseini, H.; Connolly, M.J. A comprehensive survey of wearable and wireless ECG monitoring systems for older adults. *Med. Biol. Eng. Comput.* **2013**, *51*, 485–495. [CrossRef]
24. Lara, O.D.; Labrador, M.A. A survey on human activity recognition using wearable sensors. *IEEE Commun. Surv. Tutor.* **2012**, *15*, 1192–1209. [CrossRef]
25. Al Sadawi, A.; Hassan, M.S.; Ndiaye, M. A survey on the integration of blockchain with IoT to enhance performance and eliminate challenges. *IEEE Access* **2021**, *9*, 54478–54497. [CrossRef]
26. Liang, X.; Zhao, J.; Shetty, S.; Liu, J.; Li, D. Integrating blockchain for data sharing and collaboration in mobile healthcare applications. In Proceedings of the 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), Montreal, QC, Canada, 8–13 October 2017; pp. 1–5.
27. Ometov, A.; Bardinova, Y.; Afanasyeva, A.; Masek, P.; Zhidanov, K.; Vanurin, S.; Sayfullin, M.; Shubin, V.; Komarov, M.; Bezzateev, S. An overview on blockchain for smartphones: State-of-the-art, consensus, implementation, challenges and future trends. *IEEE Access* **2020**, *8*, 103994–104015. [CrossRef]
28. Esposito, C.; De Santis, A.; Tortora, G.; Chang, H.; Choo, K.K.R. Blockchain: A panacea for healthcare cloud-based data security and privacy? *IEEE Cloud Comput.* **2018**, *5*, 31–37. [CrossRef]
29. Assiri, B.; Busch, C. Approximately opaque multi-version permissive transactional memory. In Proceedings of the 2016 45th International Conference on Parallel Processing Workshops (ICPPW), Philadelphia, PA, USA, 16–19 August 2016; pp. 393–402.
30. Arbabi, M.S.; Lal, C.; Veeraragavan, N.R.; Marijan, D.; Nygård, J.F.; Vitenberg, R. A survey on blockchain for healthcare: Challenges, benefits, and future directions. *IEEE Commun. Surv. Tutor.* **2022**, *25*, 386–424. [CrossRef]
31. Shashi, M. Leveraging Blockchain-based electronic health record systems in healthcare 4.0. *Int. J. Innov. Technol. Explor. Eng.* **2022**, *12*, 102407. [CrossRef]
32. Rahimi, N.; Gudapati, S.S.V. Emergence of blockchain technology in the healthcare and insurance industries. In *Blockchain Technology Solutions for the Security of Iot-Based Healthcare Systems*; Academic Press: Cambridge, MA, USA, 2023; pp. 167–182.

33. Ghadge, A.; Bourlakis, M.; Kamble, S.; Seuring, S. Blockchain implementation in pharmaceutical supply chains: A review and conceptual framework. *Int. J. Prod. Res.* **2023**, *61*, 6633–6651. [CrossRef]
34. Shukla, M.; Sethi, D.; Bindal, L.; Mani, K.; Upadhyay, K.; Sharma, M. Patient Monitoring System using Blockchain and IoT Technology. *Recent Adv. Electr. Electron. Eng.* **2023**, *16*, 449–459. [CrossRef]
35. Hawashin, D.; Jayaraman, R.; Salah, K.; Yaqoob, I.; Simsekler, M.C.E.; Ellahham, S. Blockchain-based management for organ donation and transplantation. *IEEE Access* **2022**, *10*, 59013–59025. [CrossRef]
36. Abdu, N.A.A.; Wang, Z. Blockchain Framework for Collaborative Clinical Trials Auditing. *Wirel. Pers. Commun.* **2023**, *132*, 39–65. [CrossRef]
37. Sharma, P.; Namasudra, S.; Chilamkurti, N.; Kim, B.G.; Gonzalez Crespo, R. Blockchain-based privacy preservation for IoT-enabled healthcare system. *ACM Trans. Sens. Netw.* **2023**, *19*, 1–17. [CrossRef]
38. Combi, C.; Pozzi, G.; Veltri, P. (Eds.) *Process Modeling and Management for Healthcare*; CRC Press: Boca Raton, FL, USA, 2017.
39. Bosanac, D.; Stevanovic, A. Trust in E-Health System and Willingness to Share Personal Health Data. In *Informatics and Technology in Clinical Care and Public Health*; IOS Press: Amsterdam, The Netherlands, 2022; pp. 256–259.
40. Calnan, M.; Ferlie, E. Analysing process in healthcare: The methodological and theoretical challenges. *Policy Politics* **2003**, *31*, 185–193. [CrossRef]
41. Samost-Williams, A.; Nanji, K.C. A systems theoretic process analysis of the medication use process in the operating room. *Anesthesiology* **2020**, *133*, 332–341. [CrossRef]
42. Lv, Z.; Qiao, L. Analysis of healthcare big data. *Future Gener. Comput. Syst.* **2020**, *109*, 103–110. [CrossRef]
43. Assiri, B.; Busch, C. Approximate count and queue objects in transactional memory. In Proceedings of the 2017 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), Lake Buena Vista, FL, USA, 29 May–2 June 2017; pp. 894–903.
44. Numan, M.; Subhan, F.; Khan, W.Z.; Assiri, B.; Armi, N. Well-organized bully leader election algorithm for distributed system. In Proceedings of the 2018 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET), Serpong, Indonesia, 1–2 November 2018; pp. 5–10.
45. Chong, Y.W.; Ismail, W.; Ko, K.; Lee, C.Y. Energy harvesting for wearable devices: A review. *IEEE Sens. J.* **2019**, *19*, 9047–9062. [CrossRef]
46. Rashid, N.; Al Faruque, M.A. Energy-efficient real-time myocardial infarction detection on wearable devices. In Proceedings of the 2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), Montreal, QC, Canada, 20–24 July 2020; pp. 4648–4651.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.