Blockchain-Enabled Secure and Efficient Data Sharing Scheme for Trust Management in Healthcare Smartphone Network

Rati Bhan Rati Bhan (ratibhan.16dp000126@cse.ism.ac.in)
Indian Institute of Technology Dhanbad
Rajendra Pamula Rajendra Pamula
Indian Institute of Technology Dhanbad
Parvez Faruki Parvez Faruki
Jyoti Gajrani Jyoti Gajrani

Research Article

Keywords:

Posted Date: September 12th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-2034814/v1

Blockchain-Enabled Secure and Efficient Data Sharing Scheme for Trust Management in Healthcare Smartphone Network

Rati Bhan[1], Rajendra Pamula[2*], Parvez Faruki[3], and Jyoti Gajrani[4]

[1]Department of Computer Science and Engineering, Indian Institute of Technology (ISM), Dhanbad, 826004, India(e-mail: ratibhan.16dp000126@cse.ism.ac.in)
[2]Department of Computer Science and Engineering, Indian Institute of Technology (ISM), Dhanbad, 826004, India
[*]rajendra@iitism.ac.in
[3]Department of Technical Education, Govt. of Gujarat, India (e-mail: parvezfaruki.kg@gmail.com)
[4]Department of Computer Science and Engineering, Engineering College Ajmer, India (e-mail: jyotigajrani@ecajmer.ac.in)

ABSTRACT

Internet of Medical Things (IoMT) is an extended genre of Internet of Things (IoT) where the *Things* collaborate to provide remote patient health monitoring, also known as the Internet of Health (IoH). Smartphones and IoMTs are expected to maintain a secure and trusted confidential patient-records exchange while managing the patient remotely. Healthcare organizations deploy Healthcare Smartphone Networks (HSN) for confidential patient data collection and sharing among smartphone users and IoMT nodes. However, the attackers gain access to confidential patient data via compromised/malicious IoMT nodes from the HSN. Additionally, attackers can compromise the entire network via the compromised nodes. This article proposes a Hyperledger blockchain-based technique to identify the compromised IoMT nodes and safeguard sensitive patient records. Furthermore, the paper proposes a Clustered Hierarchical Trust Management System (CHTMS) to block malicious nodes. In addition, the proposal employs Elliptic Curve Cryptography (ECC) to protect sensitive health records. Similarly, the proposed approach is resilient against denial of service, eclipse attacks, and terminal device failure. Finally, the evaluation results show that integrating blockchains into the HSN system improved the detection performance compared to the existing state-of-the-art. Therefore, the simulation results indicate better security and reliability when compared to conventional databases.

1 Introduction

The Internet of Things (IoT) comprises physical objects connected to the Internet via embedded sensors (e.g., camera, gyroscope, and GPS) to perceive, interact, and exchange data. Internet of Medical Things (IoMT) facilitates different medical treatment concepts in the modern era of remote healthcare via smartphones. In addition, the Covid-19 pandemic has strained healthcare, necessitating the development of tools that can help with remote treatment. Numerous good examples of the IoMT that promote remote healthcare, such as smart insulin pens, oxygen/asthma monitors, connected inhalers, and many other smartphones controlled IoMT devices for patients' treatment enabling them to manage and quickly resolve their medical needs when anything goes wrong[1]. Smartphone-compatible wearable devices, such as smart-watch and biosensors, help monitor a patient's body parameters via remote IoMT, enabling observation, diagnosis, and decision-making.

Smartphones have become ubiquitous for healthcare institutions, reducing communication cost and delay due to their sensing and app capabilities. Several studies show that smartphone apps are behind the rapid expansion of the mobile health industry[2]. Smartphones facilitate patient medical record collection, and transmission[3]. Additionally, a smartphone sensor-enabled app detects certain cardiac illnesses by monitoring and exchanging patient electrocardiogram (ECG) reports with an IoMT environment[4]. Consequently, these devices form what is known as the healthcare smartphone network (HSN)[5].

IoT software and hardware spending is expected to grow from 726 billion USD in 2019 to 1.1 trillion USD in 2023[6]. A recent Gartner, Inc. study reports that 47 percent of enterprises aim to boost their investments in the IoT despite the adverse effects of COVID-19[7]. In addition, the IoT is progressively being implemented in healthcare organizations. The IoMT assisted healthcare transformation into a more intelligent generation, offering real-time interaction and machine-to-machine connectivity[8]. It can keep track of the patients' health, provide reports to healthcare professionals, and alert them about

significant deviations. In practice, the IoT interdependent ecosystem enables multiple component communication such as real-time data collection, physical connection, point-to-point application control, data analysis, and so on[9]. The IoMT and HSN offer opportunities to improve the healthcare business. However, technological flaws are similar to conventional networks open to scrutiny[10]. According to research by Mohammed et al.[11], and Atlantic Council[12], the number of documented formation security breaches in healthcare institutions is more than double the number in other industries. It implies that safeguarding patients' privacy and sensitive data is one of the most significant issues and challenges.

Some essential characteristics of trust with diverse meanings are included in several engineering models, such as security, reliability, usability, availability, privacy, and safety[13]. For example, trust in a healthcare smartphone network is a level of confidence that deals with patient information sent from a centrally trusted server to an app or IoMT devices[14]. In addition, trust is employed to measure a node's competence in providing necessary service in wireless ad-hoc and IoT network dependability[15]. In general, creating trust in a network has several advantages, including the following:

1. Appropriate data access control depends on assessing the response of IoMT devices and their services by analyzing the trust value. However, it is very challenging to provide the same through traditional security techniques since it does not overcome such situations[16].
2. Trustworthy routing that protects critical patient information against malicious, selfish, or malfunctioning nodes can be achieved through trust[17].
3. Guaranteeing the interacting IoMT devices are trusted throughout authentication and authorization, trust makes the services' traditional security model resilient[18].

Because of clustering techniques, the trust management is a cooperative process rather than an individual duty for the IoMT node in the HSN network. Real-world WSN clustering techniques such as LEACH[19], HEED[20] , TEEN[21], and PEGASIS[22] are such examples. IoT may be installed in the kind of groups[23] which seem willing to work together to process, collect, and convey acquired data[24]. It emphasizes how various clustering techniques and group deployments allow IoTs to work together rather than independently perform their responsibilities. As a result, developing and cooperatively sustaining trust in a clustering system offers several benefits. For example, it aids the member nodes in selecting a trustworthy cluster head inside the cluster. The cluster head would become capable of similarly identifying malicious nodes(s). Furthermore, it aids in selecting trusted route neighbors via which a node transfer patient records to the cluster head in multihop clustering[20],[25].

Trust management aids in selecting trustworthy gateway nodes in the trusted route and identifying alternative trusted cluster heads by which the source node will transfer patient records to the base station (BS) in inter-cluster information exchange. Among peer-to-peer networks[26] as well as ad-hoc networks[27], a variety of trust management solutions have been suggested. To our knowledge, only a comprehensive trust management strategy, Reputation-based Framework for Sensor Networks (RFSN)[28]) presented by Ganeriwal and Srivastava, for sensor networks. Reputation-based trust management solutions[29],[30] have several drawbacks, including IoMT resource constraints. Hence, the proposed work implemented a lightweight, reliable Cluster-based Hierarchical Trust Management Scheme (CHTMS) to overcome the resource constraints issues.

1.1 Motivation

The healthcare data has private and sensitive patient records, ailment information, diagnosis data, and doctor recommendations. If the adversary gains access to the patient data, the patient data's safety, security, and privacy are compromised[31][32]. The sensitive information has a high probability of misuse for financial gain[33]. Mohammed et al.[11] studied data breaches in healthcare institutions during the COVID-19 pandemic and reported volumes of patient information leaked and misused by the supported service provider, causing resources crisis such as lack of oxygen and ventilator. Recent studies[34][35][36][37] report alarming misuse of patient health records. Hence, the security of the private records in IoMT nodes in HSN is paramount. Hence, developing an adequate trust mechanism is critical and mandatory in the IoMT and HSN.

1.2 Contributions

Traditionally trust management is handled by a single centralized server in healthcare organizations, a single point of system failure. The server may be subject to overwhelmed traffic or unexpected events. With Consideration of blockchain technology's recent growth and acceptance, it has been discovered that it offer a platform for entirely unknown parties to interact without using a trusted intermediary. Our primary focus is on thwarting insider attacks. Therefore, a blockchain-based trust management system was developed to enable HSN to protect themselves against significant insider attacks. The contributions of the paper are:

1. The proposed CHTMS algorithm is a blockchain-enabled Cluster-based Hierarchical Trust Management techniques for detecting malicious insider node. The HSN nodes examine the transactions and identify malicious node(s). We implemented lightweight, trust management approach which overcome IoT resource constraint issues. Furthermore, compared memory and communication overhead with existing state-of-the-art.

2. In comparison to typical trust management systems, which constantly concentrate on the trust values of individual nodes, CHTMS examines the trust of a cluster of IoMT devices. In addition, this technique uses less memory at each IoMT device in the network to maintain trust records.

The Proposed collaborative system combines energy-efficient data collecting with safe data exchange across IoMT nodes, using the Hyperledger blockchain. Hyperledger successfully retain a tamper-proof ledger distributed among IoMT nodes without any requirements for a trusted third central entity, ensuring data security and safety when IoMT nodes share data. The proposed approach validates security and privacy of the proposed HSN against various malicious attacks, such as the eclipse attack, Denial-of-Service (DoS), terminal device failure.

2 Background and Related Work

Section 2 discusses the HSN design and its work on preventing insider attacks, including various intrusion detection systems (IDS) and trust planning and management. This section covered the blockchain background, literature study, and other related research works.

2.1 HSN Background

Nowadays, information technology and communications system are being steadily embraced in the healthcare industry, allowing patients and healthcare specialists to communicate more efficiently. The smartphone is among the essential gadgets deployed in numerous healthcare organizations, assisting in cost reduction, data management, and outcome control. It also includes several simple applications that allow users to collect real-time data and consult with healthcare experts. For example, Guo et al.[3] unveiled a smartphone-enabled electrochemical biosensing device that enables healthcare specialists to examine patients' medical records and give accurate, personalized therapy[38]. In addition, Yang et al.[4] demonstrated an IoMT that employed smartphones to capture real-time ECG data from patients and deliver it to the appropriate healthcare institutions for review.

As a result, these Internet-enabled smartphones create a developing network platform known as the Healthcare Smartphone Network(HSN), which may be considered one form of the Internet of Things (IoT). The high-level design of HSNs is depicted in Figure 1. The former refers to smartphones used within a healthcare institution. In contrast, the latter also includes smartphones used outside the organization, the equipment patients utilize in their daily lives. Furthermore, the patients' smartphones connect with local HSNs over the Internet. So every device in the HSN is viewed as a (network) node, just like in a traditional network. HSNs, in particular, are predicted to deliver several advantages to patients and healthcare practitioners[39].

1. To deal with patients' emergency problems in a timely way.

2. Patients' financial costs and communication delays are reduced.

3. Improving healthcare organizations' resource management.

4. Maintain patient records in the secure channel against various inside security attacks.

For instance, an attacker may pretend as a patient and then attempt to hack an IoMT device in an HSN. The attacker uses the compromised device (IoMT node) to execute other attacks, including spoofing, Eclipse Attack, Vulnerability Attack, DoS, and malware distribution. As a result, develop an effective security procedure to fight insider attacks, such as recognizing malicious nodes. In addition, the system is intended to be dynamic and centralized, allowing more excellent monitoring and management.

2.2 Background on Blockchains

Due to the widespread adoption of the cryptocurrency application, blockchain technology has drawn much interest from both academic and enterprise communities. Blockchains' main goals are to make a tamper-proof transaction chain and enable payments between entities that do not have a trusted connection. As a result, blockchain has allowed mutually distrusting entities to conduct financial transactions and maintain the integrity of confidential info without depending on a central trusted third party (TTP). A blockchain typically consists of a collection of records (known as blocks) chronologically arranged using discrete-time stamps[40]. One block is linked to the preceding block using a cryptographic hashing technique; the initial block is known as the *genesis block.* A block typically contains a timestamp, payload, and cryptographic hash computed by every

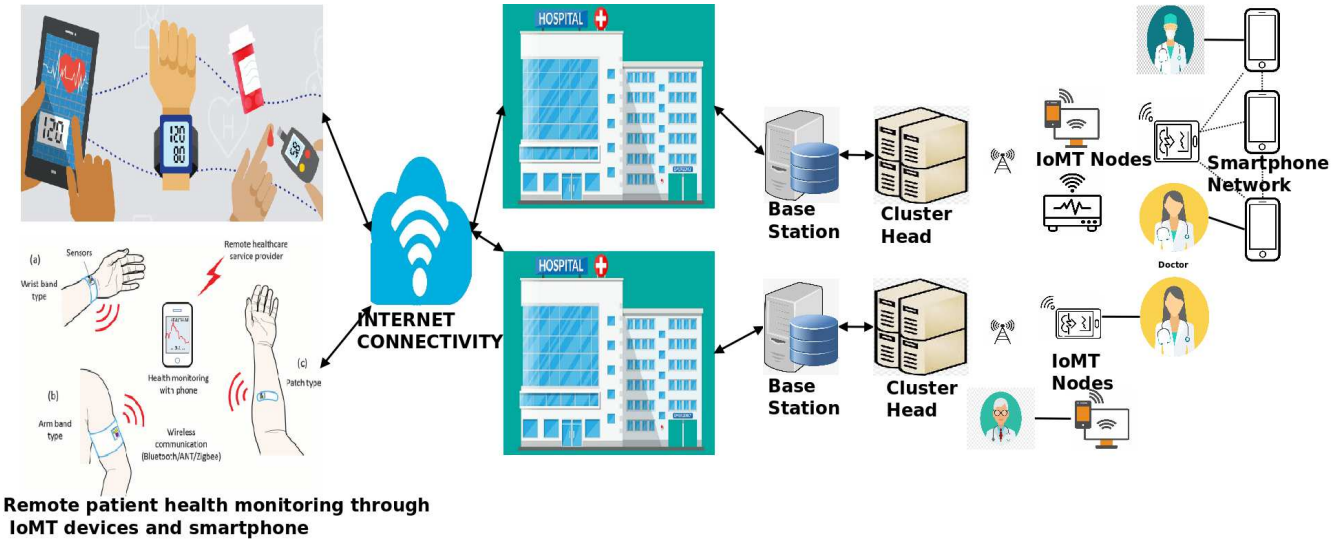

Figure 1. High-level design of HSNs

preceding block.

Role of Blockchain Blockchain technology has gotten quite a bit of attention[41]. It is a repository for distributed ledgers made up of data blocks produced through cryptography. A distributed network manages or controls a blockchain that provides traceable, transparent, and immutable data storage while ensuring data integrity. The data encapsulated in any block cannot be changed maliciously without the consent of more than half of the participants[32]. Distributed ledger, smart contracts, a consensus algorithm, and peer-to-peer connectivity are all included in a complete blockchain system. The research community explored and deployed numerous blockchain techniques across industries like economics, healthcare, the public sector (official registry), and many more[42]. Blockchain is categorized into three types explained as (i) Permissionless blockchains, also known as public blockchains, (ii) Permissioned blockchains, also known as private blockchains; and (iii) Federated blockchains, also known as Consortium blockchain[40]. The former allows anyone to participate in the blockchain as a viewer (reader) or contributor (writer) throughout the consensus process.

Permissionless Blockchains: Anyone in the permissionless blockchain network can verify the transaction and participate in the consensus process. The record of peer-to-peer transactions is kept in the block, ensuring decentralization. Before being stored in the system, each block is linked to the blockchain. The most prominent permissionless blockchain is Ethereum[43]. Complete transparency in the process and user anonymity are two significant advantages of the public blockchain network. However, using cryptographic hash techniques, it is technically possible to create a permissionless blockchain that conceals privacy-sensitive data (e.g., Zerocash[44], and Bitcoin[45].

Permissioned blockchains: It is presented as a way only a small number of readers and writers (some set of nodes only) are authorized to perform operations. The permission to access data in the private blockchain network is under strict management control. Therefore, not all the network's nodes can contribute to transaction verification and validation. However, a corporation or organization initiates, verifies, and validates each transaction. Therefore, the private blockchain is more secure, faster, and more efficient in transaction verification and validation. In comparison with public blockchains, the advantage of a private blockchain is that a corporation can pick the access privileges of individuals and allow for better privacy. Only a central competent entity (BS in our proposed approach) decides and grants to particular peers to participate in the blockchain's write and read processes. Multichain and BlockStack[46] are two of the most well-known permissioned blockchains.

Consortium blockchain: A consortium blockchain is a hybrid of public and private blockchains considered somewhat decentralized[47]. The node can choose which transaction records or data are available publicly or privately on this blockchain network. In principle, the consortium blockchain is a cross between public blockchains' low trust and private blockchains' one fully trustable entity paradigm. A consortium blockchain is primarily concerned with efficiency and transactional privacy rather than collecting and authorizing data from a single corporation or organization. R3 Corda[48] and Hyperledger Fabric[49] are well-known examples of Consortium blockchain.

Suitable blockchain for proposed approach The proposed HSN approach needed the properties of both permissionless and permissioned blockchains, so the consortium blockchain considered as most suitable blockchain. The patient and HSN network (inter base station) are connected with the consortium blockchain, while the IoMT nodes, cluster head, and base station (Intra base station) are needed permissioned blockchain. So the patient is openly allowed to join as a reader using their smartphone only. In contrast, the Base Station (BS) of the HSN network is authorized to act as a writer in a permissionless blockchain, so in this way, it maintains the integrity of the medical report. So it helps us keep patient data privacy from unauthorized persons. The Hyperledger Fabric blockchain source code partially implemented for healthcare smartphone networks is available at GitHub repository[50].

2.3 Related work

An intrusion detection system (IDS) is one of the most widely used mechanisms for securing various networks, including healthcare smartphone networks. Anomaly-based IDS and Rule-based IDS are popular types of IDS. Rule-based IDS identifies future attacks by comparing known events to stored malware signatures[51]. An anomaly-based IDS identifies a discrepancy between the main profile and a previously specified standard profile to identify malicious behavior[52]. If any possible danger is discovered, the IDS will trigger an alert.

Trust Based Intrusion Detection in Collaborative Systems: Collaborative intrusion detection is widely used in various practical contexts to increase the detection capabilities of a single IDS[53]. However, internal attacks are still a significant risk for collaborative systems. Trust management is necessary to safeguard similar networks against insider threats. The social science term *trust* is employed to evaluate a node's reputation. For example, Bradbury et al.[54] developed a sort of distributed trust across sensor nodes to detect faulty or malicious nodes and reduce their effect on applications. This technique calculates statistical trust values and selects a confidence interval on the trust reputation by evaluating the behavior of nodes.

According to Li et al.[55], most distributed IDSs rely mainly on centralized or distributed integration, rendering the communication methods unscalable. Therefore, they proposed a dynamic detector that would work with the upcoming decentralized location and routing infrastructure to solve the communication problem among nodes. Because they believed that all peers were trustworthy, their strategy was prone to insider attacks or betrayal attacks in which specific nodes became malicious unexpectedly. Alevizos et al.[56] presented a host-based IDS cooperation architecture that provides each IDS node to transmit challenges and assess the trustworthiness of everyone based on its own experience. The forgetting factor emphasizes the importance of freshly acquired expertise. Li et al.[57] discovered that not every IDS has the same degree of sensitivity in recognizing all types of intrusions and that recognition rate should depend on their signatures and applied machine learning techniques to increase detection performance. As a result, they established the concept of intrusion sensitivity and studied how well it performed in calculating trust levels for various IDS nodes. In order to increase the robustness of IDS, we developed a trust management approach depending on intrusion sensitivity[58].

Data Security for IoMT Nodes: For data security in IoT systems, Karati et al.[59] suggested a lightweight certificate-less signing (CLS) mechanism rather than using a random oracle model (ROM) and map-to-point (MTP) function. Under the difficulty of enhanced Bilinear Strong Diffie-Hellman (BSDH), the novel CLS technique is safe even against Type-I and Type-II attackers. Moreover, it is a very efficient CLS scheme widely useful for data authenticity of IoMT node in the HSN System.

3 Proposed Methodology

This section presents the proposed blockchain-based healthcare smartphone network maintaining a hierarchical trust monitoring system. The proposed HSN collaborative system combines energy-efficient data collecting with safe data exchange across IoMT nodes using the Hyperledger blockchain. The Hyperledger blockchain successfully retains a tamper-proof ledger distributed by the involved IoMTs without any requirements for a trusted third central entity, ensuring data security and integrity during IoMTs communication[6061]. In our proposed HSN, the malicious IoMT nodes are identified at the cluster level, with the help of trust computation of individual IoMT nodes and Integration of SNORT IDS to prepare a blocklist of malicious nodes. Later, analyze the proposed model's security features against various insider attacks. At last, analyze the robustness of HSN by finding the rate of transaction failure and successful undergoing DoS attacks and compare the proposed HSN with other solutions in performance and security analysis.

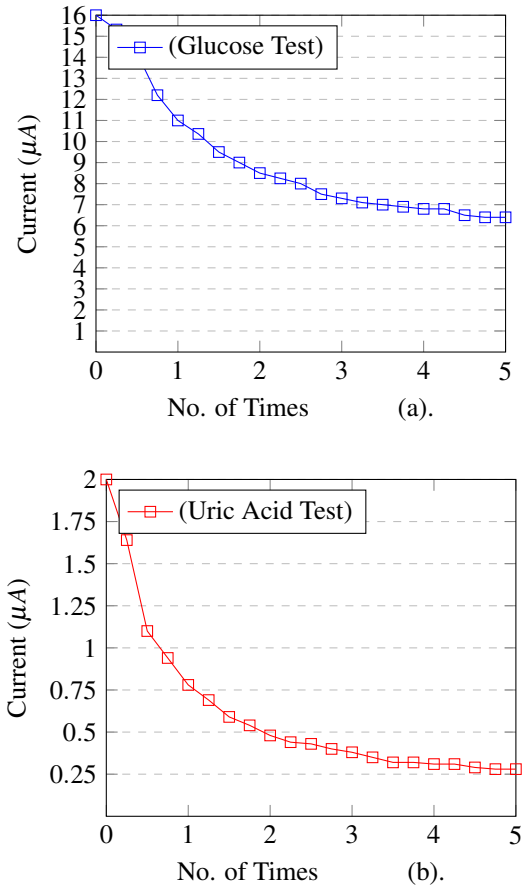

Figure 2. Chronoamperometric curves, represent the reading of Current (μA) generated during testing of (a). Bloose Glucose and (b). Uric Acid; from blood sample taken from healthy person measured by smartphone powered IoMT medical dongle.

3.1 Smartphone Powered IoMT devices

Recent studies focused on applications of smartphone-powered biomedical devices with biochemical, immunological, and genetic characterization[62]. Instant monitoring of Uric Acid (UA)[63], immunological detection of HIV[64], prostate-specific antigen (PSA)[65], and protein bio-markers demonstrate using smartphones as tiny electrochemical analyzers at the point of care. This research presented an IoMT device (Medical dongle) powered by a smartphone and functions as a miniature electrochemical analyzer to measure uric acid and blood sugar using a disposable test strip. Using OTG wire, the medical dongle stays connected to the smartphone. The biochemical data are stored on a smartphone and sent to the HSN. The purpose of the HSN platform is to accept the patient records captured by the medical dongle and deliver expert medical service. Through the built-in HSN platform, the concerned doctor will immediately access the relevant health data of their patients and remotely start prescriptions.

The hefty medical biochemical analyzer (Hitachi 7600P) use to measure the UA and blood glucose at concentrations of $401 mol/L$ and $5.1 mmol/L$, respectively. The blood samples with a range of UA/blood glucose values are used to calculate the test strip reliability. The low concentration level (for UA below $200 \mu mol/L$ and glucose below $6.0 mmol/L$), The range of medium concentration level (for UA $201 \mu mol/L$ to $435 \mu mol/L$ and for glucose $7.0 mmol/L$ to $10.0 mmol/L$), The high concentration level (for UA above $436 \mu mol/L$ and glucose above $11.0 mmol/L$). The UA and glucose test strip integrate with the medical dongle to ensure consistency and reliability. The current generated by the electrochemical reaction measurements by the IoMT device (medical dongle) using glucose and UA test strips were presented in Figure 2 as well. The current signal mapped to get summarized test value and coefficient variance in IoMT results is below 3 % and 5 %, for the glucose and UA test as shown in Table 1 and 2 respectively. These test findings showed that the smartphone-powered IoMT devices are exact for measuring UA and blood sugar has promise for use remotely in healthcare situations.

3.2 Scheme for trust management

In our proposed model, two topologies are used in the trust model.

level of Glucose	Value	Coefficient variance
Low	4.1 +/- 0.1	3.0 %
Medium	8.6 +/- 0.2	2.5 %
High	13.0 +/- 0.3	2.5 %

Table 1. Level of glucose measured by smartphone powered IoMT medical dongle

level of Uric Acid	Value	Coefficient variance
Low	165 +/- 6	3.0 %
Medium	287 +/- 7	2.0 %
High	570 +/- 10	2.0 %

Table 2. Level of uric acid measured by smartphone powered IoMT medical dongle

1. The first is the intragroup topology, in which centralized trust management takes place inside the hospital.
2. The other is intergroup topology, in which distributed trust management occurs among the hospitals.

Each IoMT node (i.e., IoMT devices and smartphones) in the HSN network calculates the trust values for every other node (only for group members) separately. Then, a node assigns one of the three possible state options based on the trust values. The possible states include 1) trusted nodes, 2) untrusted nodes, or 3) uncertain nodes. For mathematics brevity and to provide an appropriate level of granularity, the 3-state approach was chosen. Following that, each node transmits the trust state of all the group nodes to the Cluster Head (CH) installed at the individual Departmental Information Service Center. After that, centralized trust management takes control. The CH recognizes the malicious node(s) and alerts the Base Station (BS) located at the Hospital Information Service Center of the same and receives the trust statuses of every group member.
Additionally, upon request from the BS, each CH sends the trust values of the neighboring CHs. As soon as the BS has all the data, it assigns one of the group's three potential states. Upon request from a particular CH, the BS will additionally send the present condition of certain CHs. Our grouped trust management model is composed of three main phases: (i). Computation of trust first at the node level, (ii). computation of trust at the level of cluster head, and (iii). last calculation of trust at the base station level.

3.2.1 Computation of Trust at the Node Level

In HSN, the node may be any IoMT devices or smartphones. Peer recommendations or time-based prior contact at the node level are used to calculate a trust value. For instance, when node A wants to interact with node B, it first checks to see if A has any prior history of interacting with B within a particular period. If the answer is yes, node A makes the right decision based on the last communication; if the answer is no, node A uses the peer recommendation method.

Trust Evaluation based on Past communication Using a Time-Based Approach The trust calculation indicates the degree of confidence in the node's reliability at each node. As a result, the authors introduce the concept of a sliding time window, which takes relative time into account and mitigates the effect of network conditions on the cumulative trust calculation because the communication protocol is often preceded by timestamps, every node that delays packet delivery by utilizing the sliding timing window.

The timing window (Δt) divided into multiple time units determines previous communication success and failure rates. Prior communication takes place at each point in time. It logged the units contained within the timing window. When a unit of time passes, the window slides to the right by a one-time slot. As a result, the window keeps forgetting the experiences as time passes. The window length could be reduced or increased based on scenarios derived from network analysis. The figure 3 illustrates the HTMS time window scheme. The time window Δt is made up of four units. The initial unit of (Δt_1), indicating the number of success and failure communication/transactions are two and one, respectively. Throughout the entire Δt_1 interval, the number of success and failure attempts is ten and four. Following the passage, Δt_1, the new time interval Δt_2 decreases the values associated with prior communication that occurred during the very first unit Δt_1 (S=2, F=1) and considers only the values of the last three units of (Δt_1) plus the value of a recently added unit (S=5, F=2) on the right.

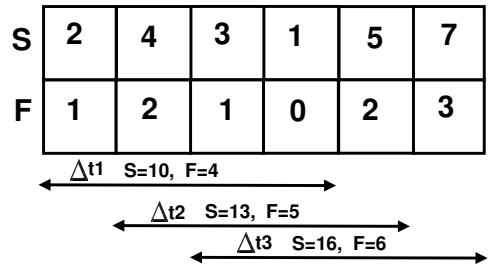

Figure 3. Sliding time window based scheme for trust computation among HSN network entities.

$$T_{A,B} = \Big[\, 100\Big(\frac{S_{A,B}}{S_{A,B}+F_{A,B}}\Big)\Big(1-\frac{1}{S_{A,B}+1}\Big)\Big] \\ = \Big[\, \frac{100(S_{A,B})^2}{(S_{A,B}+F_{A,B})(S_{A,B}+1)}\Big] \tag{1}$$

Along with the time window based on past communication, the trust value $T_{A,B}$ of node B to node A, the range defined as 0 to 100 (took the nearest integer), $S_{A,B}$ is the weighted sum of the number of successful communication between node A and B during time Δt, and $F_{A,B}$is the total number of failed communication between node A and B during Δt time. The expression $(1-\frac{1}{S_{x,y}+1})$ represent as the number of successful communication increases. This method was chosen over a linear function because a linear function very slowly reaches one as the number of successful interactions increases; thus, it would take significantly longer for a node to increase its trust value for another node. Likewise, it takes a long time to decrease the trust value of a node in the same cluster formed at each departmental level. Thus, the equation 1 is designed to mitigate the effects of a declaration of a few wrong communications caused by network traffic problems. The trust value rises when the number of successful intercommunication increases but remains modest if communication failure is also relatively high. For instance, if six unsuccessful interactions and five successful, the trust value is 4.5. Following is the calculation of the trust value to judge the trust state of a node (N_j).

$$N_j(T_{A,B}) = \left\{ \begin{array}{ll} trusted & 100-C1 \le T_{A,B} \le 100 \\ uncertain & 50-C2 \le T_{A,B} < 100-C1 \\ untrusted & 0 \le T_{A,B} < 50-C2 \end{array} \right\}.$$

Where C1 denotes half of all trusted node average values, and C2 represents one-third of all untrusted node average values. The robustness of a node is directly affected by the use of half and one-third of average values in assessment.

$$C1 = \Big[\, \frac{1}{2}\Big(\frac{\sum_{i\in X_a} T_{a,i}}{|X_a|}\Big)\Big] \qquad C2 = \Big[\, \frac{1}{3}\Big(\frac{\sum_{i\in Y_a} T_{a,i}}{|Y_a|}\Big)\Big] \tag{2}$$

X_a is the set of trustworthy nodes in the list of node A, and Y_a is the set of untrustworthy nodes in the list of node A. N represents the total containing trustworthy, untrustworthy, and unclear nodes. Initially, all nodes' trust values are 50 while f and g are set to 20 and 15, respectively, to maintain the constraint: f>g>1. It helps to avoid ambiguous zone between trusted and untrusted Zones.

Evaluation of trust value based on peer recommendations Let us say a group consists of n nodes. In addition, every node keeps a trust value for all the other nodes in a group. When a node needs peer recommendation, it requests all group member nodes, excluding untrusted ones. For example, Node M wants to compute the trust value of Node N, where |X| is the total number of trusted nodes in a group. The new trust value for node N is calculated as:

$$T_{m,n} = \Big[\, \frac{\sum_{i\in X_a} T_{m,i} * T_{i,n}}{100 * |X|}\Big] \tag{3}$$

Where $T_{m,i}$ is the recommender's trust value, It is also considered as a recommender's weighted value multiplied by the trust value sent by all the recommenders. While $T_{i,n}$ is the trust value sent by all the trusted nodes for node N. Node N trust value shall not exceed the range of trust value among both node N and all the recommender nodes of the group X.

3.2.2 Computation of Trust at the Cluster Head Level

Each node transmits the trust state of all the group nodes to the Cluster Head installed at the Departmental Information Service Center. Authors presume the CH has more computing power and memory.

Computation of CH Group's Trust State CH requests nodes for the trust states of all other group members to determine the overall trust value of nodes in a cluster. For two reasons, the trust states employed rather than actual trust values. First, the communication cost is reduced because only a simple state is conveyed to the CH. Second, a particular node's trust constraints differ from other nodes. For example, a given trust value may belong to the trustful zone for one node. Although for other nodes, it may only belong to the uncertain zone. As a result, calculating the global trust status of a collection of nodes are more practical and efficient if used the trust state. For example, let us say the group contains n nodes and one CH. The CH will transmit the request packet to the group regularly. As a result, everyone in the group conveyed their computed trust status (S) to the CH. The trust state variable (s) can be in three states: trusted, untrusted and uncertain. All these trust state values will be maintained by the CH in the form of a matrix as given as:

$$M_{ts} = \begin{bmatrix} S_{ch,1} & S_{1,1} & \dots & S_{1,n} \\ S_{ch,2} & S_{2,1} & \dots & S_{2,n} \\ \dots & \dots & \dots & \dots \\ S_{ch,n} & S_{n,1} & \dots & S_{n,n} \end{bmatrix}$$

Where M_{ts} denotes the matrix of the cluster head's trust state, the trust state $S_{ch,1}$ represents the trust state of node one at the cluster head, while $S_{1,1}$ to $S_{1,n}$ represent the trust state of node one sent by the remaining n nodes in a group. Based on the computation of relative difference in the represented matrix of trust states of the cluster head (M_{ts}), assigned a global trust status to a node n, i.e., $S_{ch,n}$. As a result, the cluster head declares a random variable R, S_n is the sum of n random variables, and $\sqrt{\frac{n}{3}}$ is the standard deviation. The cluster head defines the behavior of random variable as follow:

$$R(S_{a,b}) = \left\{ \begin{array}{lll} trusted & when & S_{a,b} > 1 \\ uncertain & when & S_{a,b} = 1 \\ untrusted & when & S_{a,b} < 1 \end{array} \right\}.$$

Computation of Inter-Group Trust State During inter-group communication, every cluster head keeps track of their previous interactions between other groups in the same ways as individual nodes maintain track of all other nodes. A group's trust values are computed based on previous interactions or information from the base station. Assume cluster head A wishes to determine the trust value $T_{a,b}$ with another cluster head B. It determined trust value either using a time window based previous transactions $PT_{a,b}$ (when $PT_{a,b} \neq \phi$) or a suggestion from their base station $BSS_{a,b}$ (when $PT_{a,b} = \phi$), the value of $T_{a,b}$ calculated as given below:

$$T_{a,b} = \left\{ \begin{array}{ll} \left[\frac{100(S_{a,b})^2}{(S_{a,b}+F_{a,b})(S_{a,b}+1)} \right] & if \quad PT_{a,b} \neq \phi \\ BSS_{a,b} & if \quad PT_{a,b} = \phi \end{array} \right\}.$$

3.2.3 Computation of Trust at the Base Station Level

The base station keeps a record of previous interactions with Cluster heads in the same way as every node does, and the base station periodically computes the trust value between every cluster head TV_{bs,ch_i} during the time window Δt as:

$$TV_{bs,ch} = \left[\frac{100(S_{bs,ch})^2}{(S_{bs,ch}+F_{bs,ch})(S_{bs,ch}+1)} \right] \tag{4}$$

The total number of successful conversations between Base Station and Cluster Head during Δt time is $S_{bs,ch}$. In contrast, the total number of failure interactions between Base Station and Cluster Head during Δttime is $F_{bs,ch}$. The authors assume that the network contains G_n groups. The Base Station broadcasts the request packets to all the Cluster Heads regularly. In response to Base station requests, all cluster heads send their computed trust state vector TSV_{ch_i} to the Base Station. Now Base Station prepares a consolidated trust state vector and computes group-wise (G_i) trust state TSV_{bs,g_i} as given below:

$$TVS_{ch} = (TVS_{ch_1} + TVS_{ch_2} .... + TVS_{ch_n})$$

$$TSV_{bs,g_i} = \left[\frac{\sum_{i=1}^{G_n-1}(TV_{bs,ch_i})(TV_{G_i,G_j})}{G_n - 1} \right] \tag{5}$$

In the equation 5, $TV_{bs,ch}$ is the trust value between Base Station and Cluster Head, TV_{g_i,g_j} is the trust value between different two groups, namely G_i and G_j, while G_n represent the total number of groups in HSN.

3.3 Trust Measurement on Private Blockchain-Based Data Transmission

Security researchers have received much attention on keeping connected medical equipment trustworthy. Medical devices are integrated with security procedures to ensure their protection. For instance, if an IoMT device in HSN is compromised, attackers can use the infected device to exploit additional IoMT devices. Insider attacks are a significant issue for IoMT and HSN due to their distributed architecture. As previously stated, trust-based interaction is a crucial security strategy for preventing insider attacks. These methods rely on a central server to oversee the trust estimation process and decide. However, most IT people in healthcare institutions are not security experts. In such a scenario, a central server could be a weak point to the failure of the entire system. In short, it is not easy to know whether users trust the central server regarding security. With the rise in popularity and use of blockchain technology, advanced research started into the combination of trust management and blockchains. It is because blockchain technology allows unknown (or untrusted) nodes to exchange data in a verifiable way without the requirement of any trusted intermediary [28], [52]. Our objective throughout this paper is to propose a trust management system based on blockchain for securing the healthcare industry from insider attacks.

3.3.1 Integration of SNORT IDS in HSN IoMT Node

IDS technique includes sophisticated interaction between several HSN nodes and a single central server, as shown in Figure 4[5]. More precisely, place lightweight IDS (SNORT mobile version) on the phones to aid network monitoring, traffic recording, and security policy enforcement. There are usually three parts: a communication component, a traffic monitor, and a blocklist.

1. Communication component: This component is incapable of connecting to and transmitting essential information to a central server, which is essential to the interaction's success. It also assists in updating its blocklist depending on information received from the server.

2. Traffic Monitor: It is required to inspect traffic, transfer data, and record data to the communication component.

3. Blocklist: Most components include a list of the prohibited smartphone or HSN nodes based on trust values computed on the server-side. The list is planned to be dynamic based on healthcare management's response to decreasing the effect of false positives[5].

3.3.2 Trust Measurement at Centralized Base Station

The base station is responsible for assessing trust at a different level in the HSN system and malicious node detection. It generally has three key components: (i) communication component, (ii) trust computation component, and (iii) Allowlisted and blocklisted HSN nodes.

1. Communication Component: This component is managed connections between nodes and the server. It assists in gathering essential data from nodes to aid in the trust computation process and forwarding the revised blocklist to the relevant HSN nodes.

2. Trust Computation Component: This component primarily assists in calculating HSN node trust levels based on the obtained data and peer recommendation, identifying malicious nodes, and constructing the blocklist.

3. Allowlisted and Blocklisted HSN Nodes: It keeps the updated list of blocklisted and allowlisted nodes. Some security measures, in particular, are used to ensure that the list is updated dynamically and correctly.

3.3.3 Trust Management Based on Blockchain

Healthcare management demands the design mentioned in figure 1 with a single central server; however, it could become a weak point of failure. Because blockchains allow multiple nodes to interact in a distributed fashion with no need for a central authority, blockchain is becoming popular gradually. By merging blockchains, proposed a trust management system. The blockchain-enabled trust management strategy is depicted in Figure 5, which divides the HSN into two essential layers: the HSN and the blockchain.

1. HSN Layer: This feature enables HSN nodes to communicate with the centralized base server in the usual way. It helps a healthcare institution keep its existing foundation while lowering the cost of deployment. There are alternative approaches to deploying blockchain-enabled trust management, but it needs modification in the overall design.

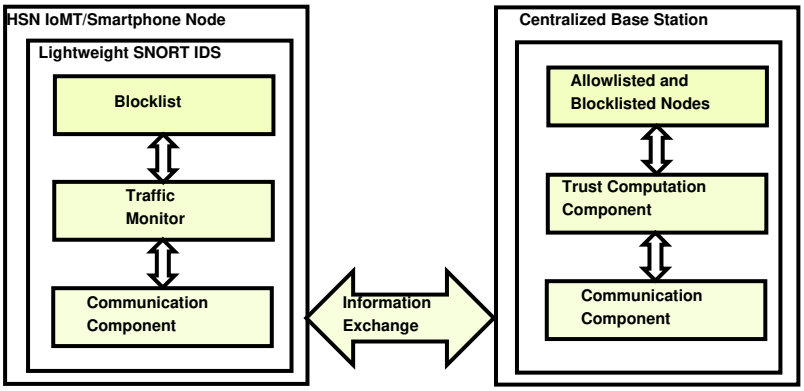

Figure 4. Typical Lightweight SNORT IDS Communication between HSN Node (IoMT/Smartphone) and Centralized Base Station

2. Blockchain layer: This layer creates a private blockchain that permits each node to submit undesirable or malicious packet characteristics. Because each node has accessibility to the chain and therefore can examine the aspects of malicious packets, the node immediately updates its local blocklist, gain further information, and transmit messages to the target node. At the same time, it is challenging to update the list quickly in traditional architecture.

The blockchain-enabled trust management strategy has two advantages: 1) By monitoring blockchains, it assists in the rapid update of blocklist among HSN nodes. 2) It permits some more capable nodes to communicate to potentially anomalous nodes and learn more about their traffic status by checking blockchains. Furthermore, per the trust computation equations 2, a malicious node is discovered quickly by lowering the threshold value.

3.3.4 Trust Measurement-Based Threshold Exploration

$T_{a,b}$ represent a trust value between node a to node b, calculated using the equations 3. However, an HSN node stopped for a malicious packet, leading to a high false-positive rate produced by accidents or careless operations. As a result, the dynamic blocklist design method reduces the impact of mistake rates. For example, the following judgment with a $T_{a,1}$ threshold:

1. If the trust value of blocked node $T_{a,b}$ is less than threshold T, it withdrew from the blocklist.

2. otherwise, node kept on the blocklist.

3.4 Blockchain-Based Secured Data transmission Among HSN IoMT-Nodes

The IoMT-based data collecting and outputs strategy is based on storage and query requests, which feed inputs to the blockchain-enabled storage system after each cycle of data gathering. IoMT encrypts gathered data through its private key and submits it to the blockchain as a storage request since each IoMT has a legitimate connection with it. IoMTs query the blockchain for data sharing, and the blockchain return the results. We create a centralized, safe, and trustworthy storage system that establishes consensus on a trustless ground and ensures that data/records are never tampered.

Hyperledger provides a more robust ecosystem than Ethereum and allows the development of smart contracts and business logic to be relatively straightforward. Furthermore, whereas the Ethereum platform is a public blockchain for open-source applications, Hyperledger is a private consortium blockchain meant for storing sensitive data, making it more appropriate for our experiment. Our simulation environment is formed in Python, and Hyperledger has already offered a Python interface for communicating with the blockchain. The Hyperledger blockchain meets our needs and is utilized to build the blockchain network. In Algorithm 1, employ pseudocode to describe the safe data exchange process across IoMT nodes.

Prepare the genesis block configuration file during the initialization process, which defines a separate blockchain network ID. In hexadecimal, the difficulty is initialized to 0 x 0400 (1,024 in a decimal number system). 1/1024 means that, on average, 1 out of every 1,024 hash computations will be successful. Reset the gasLimit to 0xFFFFFFFF, which is the maximum number of calculations a block can handle. The coinbase parameter of the 160-bit address receives either rewards upon successful

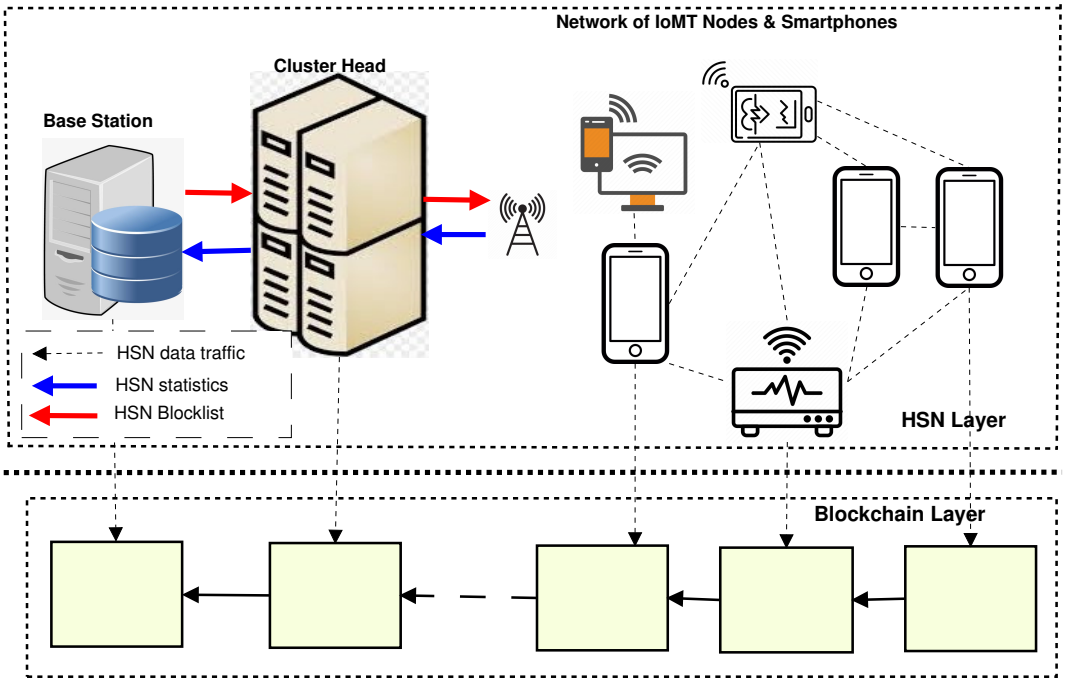

Figure 5. HSN Trust Management Based on Layered Blockchain

mining. The mixhash and nonce parameters are 256-bit and 64-bit hash values, respectively. These are used in tandem to determine if a block has been cryptographically mined and is legitimate. The parent hash parameter is the Keccak 256-bit hash value of the parent's full block header. It has both nonce and mixhash in it. The timestamp parameter provided at the start of the block returns the Ubuntu OS time(t) output. The timestamp aids in the verification of the chain's block order. At last extraData parameter is a 32-byte additional space that can be used as needed. In the private chain, subsequently create numerous Hyperledger nodes.

Each Hyperledger node launched will start a Javascript Console, and built-in objects execute tasks like mining, transactions, and querying blocks. Then Solidity programming language is used to create a smart contract that contains the initialization of the IoMT node accounts, including the storage and query functions for the data gathered. The smart contract is being deployed on the blockchain, in which its information is available to all Hyperledger nodes and interacts with each other, so it needs a storage system to share data easily. Any IoMT with blockchain access has unrestricted access to all data records. Although, It is expected that the data kept on the blockchain cannot be recorded indiscriminately since it leads to malicious IoMT nodes storing fraudulent and forged data. Hyperledger controllers can check data quality and establish a Central Authority to oversee the validity and data ownership of all IoMTs and maintain their public keys. The Certificate Authority (CA) of Hyperledger Fabric offers the following features:

1. To user registry CA connects with LDAP or registers identities.
2. Enrollment Certificates are issued (ECerts).
3. Certificate renewing and cancellation.

There are two ways of interacting with a Hyperledger Fabric CA server: via the Hyperledger Fabric CA client or through one of the Fabric SDKs. Therefore, communicate with the CA server in one of two ways: (i)using the CA client, or (ii) using the Fabric SDKs. The CA client implements at Cluster Head(CH) level.

Algorithm 1 depicted step by step process of workflow in our proposed approach to the healthcare system. Deploy the Hyperledger blockchain in the HSN network and design a hyperledger block for each node. Later deploy *smart contract* in blockchain and start the block mining process. From steps 4 to 7 in algorithm, use modern asymmetric key cryptography named Elliptic Curve Cryptography (ECC) to calculate public key (puK_i) and private key (prK_i) for each IoMT node(i) from set $\mathscr{I}^{66}$. Because ECC employs fewer keys and signatures for almost the same degree of security as RSA and enables rapid key

Algorithm 1: Blockchain-Based Secured Data transmission Amoung HSN IoMT-Nodes

Setup a private blockchain and deploy N Hyperledger blocks;
Initialize a *smart contract* on the Hyperledger blockchain;
Start *block mining*;
$\mathscr{I} \leftarrow$ set of HSN IoMT nodes;
for each $IoMT_i$ in $\mathscr{I}$ do
    Define *ECC_Curve(a, b, SubGroup(p, g, n, h), secp256k1)* to calculate (puK_i, prK_i)
    BS store $keyRecords(i, puK_i)$
end for
for each $IoMT_i$ in $\mathscr{I}$ do
    $d_i^t \leftarrow$ data collected from $IoMT_i$ at timestamp t
    Compute cryptographic hash on collected data $H(d_i^t) \leftarrow SHA256(d_i^t)$
    $Dsign_i \leftarrow dsignECDSAsecp256k1(H(d_i^t), prK_i)$
    Send $(d_i^t, Dsign_i)$ to the BS.
    if $verifyECDSAsecp256k1(d_i^t, Dsign_i, puK_i)$ is valid **then**
        if $IoMT_i$ is trusted **then**
            $Dsign_{BS} \leftarrow dsignECDSAsecp256k1(H(d_i^t), prK_{BS})$
            $Return(d_i^t, Dsign_{BS})$ to $IoMT_i$
        end if
    end if
    if $IoMT_i$ receives digital signature($Dsign_{BS}$) from BS **then**
        Initiate blockchain $transaction(d_i^t, Dsign_{BS}, puK_i)$
        if $verifyECDSAsecp256k1(d_i^t, Dsign_{BS}, puK_{BS})$ is valid **then**
            Wait for majority confirmation and *commit transaction* to blockchain
        end if
    end if
end for

generation, key agreement, and digital signatures. It is considered that ECC cryptography is a natural contemporary successor of the RSA cryptosystem. As per a comparison study by Zeinab et al.[67], the smaller key size of 256 bits ECC algorithms equivalently secure as RSA algorithm with a wide key size of 3072 bits, so lightweight ECC algorithm much secure and easily computed by low computing power devices. In this way, the ECC algorithm efficiently runs on each IoMT device. The base station (BS) store the public key (puK_i) of each IoMT node(i).

After calculating public key (puK_i) and private key (prK_i) for each IoMT node(i), In steps 9 to 12 in algorithm, We collect the temporal data (d_i^t) from each IoMT node(i) and compute cryptographic hash value using Secure Hash Algorithm (SHA) version 2 and produce 256 bit hash digest value $H(d_i^t)$. The IoMT nodes compute digital signature $Dsign_i$ using Elliptic Curve Digital Signature Algorithm(ECDSA) secp256k1 curve along with the private key (prK_i). The IoMT node send data(d_i^t) and digital signature($Dsign_i$) to the Base Station(BS). From steps 13 to 16 in algorithm, BS verify digital signature($Dsign_i$) using public key of IoMT node(puK_i). if digital signature($Dsign_i$) is valid, then BS also check the level of trust based on their previous communication, if IoMT node(i) is trusted, then BS compute digital signature($Dsign_{BS}$) for hash value($H(d_i^t)$) of data(d_i) using BS private key(prK_{BS}) and return it back to the requested IoMT node. At the last, the algorithm steps 19 to 22, IoMT node initiate the transaction using digital signature of BS($Dsign_{BS}$) along with data(d_i) and public key(puK_i). At last non-mining nodes verify and validate the transaction and commit it when they get confirmation from a sufficient number of non-mining nodes.

3.5 Security Against Insider Attacks

The robustness of the trust management protocol in a hierarchical healthcare system against insider attackers is examined in Section 4. Our premise is that benign nodes often communicate successfully and submit genuine suggestions. On the other hand, Malicious nodes attempt as many failed contacts as possible and make fake suggestions regarding benign nodes. The distinction between benign and malicious nodes is arbitrary. From the perspective of one node, a node may be benign, but from the perspective of another IoMT node, it may be malicious. Section 3 focused on capturing the activity of malicious nodes and simulating how they try to gain an unfair benefit in the trust model. Then, demonstrate the robustness of the protocol to face such malicious activity. We have applied this analysis to higher clusters in a modular manner. Therefore, we need to start with malicious and unfair advantage concepts. Both of these characteristics define a malicious node. A malicious node's purpose while dealing with other nodes is to have as many failed interactions as possible to retain the following objectives:

- Achieves a more considerable trust value than the actual computed trust value; more crucially, enters the trusted zone while its actual position is in the untrustworthy or uncertain zone.
- It tries to lower the trust value of the benign node, and
- If feasible, raise the trust value of a cooperating malicious node.

Even though blockchain security has been continually improved, attackers have a variety of techniques to hack it[68]. The authors discuss several possible attacks in our proposed system and offer solutions in this section.

3.5.1 Security Resilience Analysis of Trust Management System

Describing the pre-defined objective of malicious nodes shows that the trust management system is resilient to the nodes, cluster, and base station level if it can prevent malicious nodes from achieving their goals. Unfortunately, devising a plan to stop such activity is difficult entirely. However, to secure the proposed HSN system, somehow quantify the boundaries of such nodes. This guarantee assures that an intelligent node attempting to decrease the number of successful transactions with other nodes while remaining in the trusted zone would be unable to achieve its objectives except within defined bounds. More specifically, the intelligent node must keep the number of successful transactions greater than or equal to the number of failed interactions.

3.5.2 Transaction Forgery by HSN Nodes

The data kept on the blockchain is often comprehensive, correct, and consistent. An attacker who attempts to tamper the persistent data will have to pay a high price. As a result, it is more prone to attack while transferring the data requests. Furthermore, it can intercept transactions posted by the IoMT node to the blockchain. The attacker can impersonate the trusted nodes and submit phony transaction requests to the blockchain network after often accessing the encrypted transaction contents.

In the proposed scenario, each HSN node gathers data and makes storage requests as transaction data to store in the blockchain; however, the attacker can intercept the request and manipulate the amount of data gathered to be false or incorrect. It is conceivable that the attacker's HSN node is an authentic device inside the blockchain network while it tries to inflate

the quantity of data acquired by transmitting false data while requesting storage. The computation of the private key of CA is difficult or impossible for an attacker node. Thus, it cannot fabricate the confirmation message of CA. Instead, the CA will authenticate whether the communication originates from a legitimate device once the attacker delivers the forged storage request. If such a case, verify if the node genuinely holds the data specified in the request. CA will append the digital signature to the valid transaction and send it back to the node if everything checks out. When the transaction proposal is received, the node uploads the transaction data storage to the blockchain network, which would also check whether it is the actual digital signature of the CA or not.

3.5.3 Eclipse Attack in Blockchain Network

The Eclipse attack on a blockchain network in which the attacker effectively takes control of peer-to-peer network[69]. In an Eclipse Attack, the attacker tries to get control of the node of the intended victim. Every Ethereum node in a blockchain network depends on connections to many other nodes to get a comprehensive network view. Therefore, an attacker prohibits the victim from obtaining complete information about the network.

In our proposed scenario, an attacker employs an Eclipse Attack to prohibit non-mining nodes from accepting query and storage requests to the blockchain and make it challenging to initiate additional smart-contract events, disrupting regular device-to-blockchain interaction. Only two malicious blockchain nodes are required to isolate and affect other nodes for an eclipse attack. The software upgraded version of the blockchain addresses this vulnerability and stops increasing the number of malicious nodes to launch an attack.[7071].

3.5.4 Vulnerability Attack by Smart Contact Users

A blockchain platform with open source smart-contract capability provides visibility to each user. Therefore, it is elementary for attackers to expose a significant flaw in a smart contract. It is presumed that an attacker may use smart-contract flaws to tamper with the data that persist in the blockchain, and if a more significant vulnerability exists, the blockchain's data is stolen and deleted. Therefore, to eliminate the exposure of recursive calling[72], timestamp reliance, arithmetic challenge[73], and return value issue[74], We conducted security tests and code audits while building the smart contract.

3.5.5 Majority attack by Malicious Users

When most nodes in a blockchain network turn malicious and cooperate with the attacker, or when the attacker controls more than 50% of the processing power in the blockchain network. The blockchain data is not assured of the dependable security[75]. The attacker can use the computational power to tamper with blockchain records. The blockchain majority attacks are prevented by being designed as a private chain in the proposed HSN blockchain network where the system owns and controls the mining nodes. The mining nodes are truthful, obey the rules to some extent, and ensure that the attacker can not gain more than half of the processing power, ensuring that such an assault is prevented as feasible.

3.5.6 IoMT Nodes Failure

The IoMT node consistency is challenging in the proposed HSN network scenario, where each IoMT node's behavior might be significantly different while collecting and exchanging data dispersedly. The HSN protocol provides a set of operations for a group of nodes, ensuring that each node will accept the result of the processing. Nevertheless, over long-term usage of such IoMT devices, someone might develop software or hardware issues that prohibit them from functioning correctly, like network service failure, device downtime, and network communication failure, among other things. There are two sorts of difficulties in this category. The first is that data is lost or delayed. The other is considerably more problematic because malicious nodes in the network can communicate fake data to other users, referred to as *The Byzantine Generals Problem*[76]. In other circumstances, if an attacker knows about the devices that have failed in this scenario, it poses as an authorized or lawful IoMT and then fraudulently reports the absolute quantity of data, thus causing data insecurity or inconsistency. Blockchain consensus algorithms, such as Paxos and PBFT[77] are also used in the proposed HSN to successfully tackle the IoMT device malfunction and avoid or solve these kinds of potential difficulties.

3.5.7 Security Resilience Analysis

To examine the proposed CHTMS scheme's resilience to attacks, divide the IoMT nodes into good and compromised nodes. It is a presumption that good nodes communicate successfully and always provide genuine recommendations. On the other hand, compromised nodes strive to engage in as much unsuccessful communication as possible while also sending false recommendations regarding good nodes. The distinction between good and compromised nodes is subjective. A node may be an exemplary node in the perception of one node but a compromised node in the perception of another.

To examine this idea more precisely, record the behavior of compromised nodes, and simulate how they try to gain an unfair advantage in the CHTMS trust model. The trust management approach in CHTMS resilience to malicious behavior of a

compromised node, the same analysis deployed to a higher level in cluster hierarchy. The malicious node proceeds with the concepts of unfair benefit and compromised behavior. When dealing with other good nodes in the same cluster, a malicious node is to have as much failed communication as possible while maintaining the specific goals:

1. Keep itself high trust value than the original computed trust values; more significantly, keep itself in the trusted zone while its proper location is in the unsure or untrustworthy zone.
2. Good node's trust value decreased and forwarded to neighbor and upper level.
3. The trust value of the compromised node is always kept high.

The trust management approach at the node level seems resilient over malicious nodes when it prevents compromised nodes from achieving their pre-specified objectives. To get security for the HSN, specify the boundaries of such compromised nodes. More specifically, the trusted node must keep the number of successful communication equal to or greater than failed communication.

4 Result Analysis

We configured the experimental environment for blockchain and the comparison database MySQL. The proposed solution for collecting data by IoMTs is explained in Section 3.4 and transferring data storage queries back and forth by Java. In Section 3.4, explain private blockchain functioning in a healthcare smartphone network and propose a blockchain-based data-sharing approach using Hyperledger Fabric v2.2, which is running on Ubuntu 20.04.3 LTS for $x86\_64$ architecture, Intel Core 3.40 GHz $i7-6700$ CPU and 16 GB of RAM, with all Fabric client nodes operating on the same computer. Fabric Peer v2.2 will generate a Java 11 Virtual Machine(VM) with the $x86\_64$ architecture (OpenJDK version 11.04 11). JVMs, docker images, and other supported tools are only tested on $x86\_64$ standard architecture. Set the $CORE\_CHAINCODE\_JAVA\_RUNTIME$ environment variable to the base address of the docker image. The Java Libraries (version v2.1.0) will establish a connection to the peer while operating. It is alluded to as 'Fabric Peer Connectivity.' Execute the python script that installs Binaries, Samples, and Docker Images in the same system to install all the prerequisites. Then, run the test-network script to verify Docker Desktop 2.5.0.1. Because the Fabric CA server is not part of a cluster, SQLite was chosen as the default database with the `fabric-ca-server.DB` file is located in the CA server's root folder(home directory). The Certificate Signing Request (CSR) is configured to create Elliptic Curve X.509 certificates and keys (ECDSA). CA server allows to enroll and register more identities. The CA server is configured using LDAP and must have one already registered bootstrap identity to connect with LDAP server. Now the CA server starts listening to port 7054. Three Fabric nodes are included in our blockchain experiment, where one node accepts query and storage requests from IoMT nodes, and the remaining two nodes have the capacity of two mining threads.

4.1 Analysis of Trust Value in Normal and Malicious Condition

The involvement of trust in healthcare organizations primarily ought to analyze the efficacy of the proposed trust management system. The Sensor Network Simulator and Emulator (SENSE)[78] is used for emulation and a simulation for HSN networks to analyze the trust value in normal and malicious conditions. We implement the rule-based open-source, lightweight version of mobile Snort IDS in HSN. This study compares and contrasts two similar trust management schemes: Duma et al.[79], and Fung et al.[80]. The former used a trust-aware algorithm and intelligent trust management system to recognize insider malicious IoMT nodes.

4.1.1 The Trust Value in Normal Condition

Foremost, this research examines network traffic in its normal state. The aggregate trust levels of HSN IoMT nodes are shown in Figure 6. Set the value of the forgetting factor to 0.8 per research based on its impact analysis by Shaikh et al.[81]. Once the central server finished the data collection, the trust values seemed to become stable, i.e., extremely near to one, after some time. However, achieving one according to the proposed algorithm 1 is extremely difficult due to communication delays and insider DoS attacks.

4.1.2 The Trust Value in Malicious Condition

The analysis aims to see how well our blockchain-enabled trust management strategy performs in malicious conditions. We randomly chose three IoMT nodes to transmit malicious packets to other IoMT nodes in HSN to initiate internal attacks. The malicious traffic provoked by DoS assaults software can transmit various modified packets. Adjust the forgetting factor to 0.8 as per research based on its impact analysis by Shaikh et al.[81].

Comparison of trust value with other trust management schemes We compare the trust value of our proposed solution with two comparable trust management methods (Duma et al.[79], and Fung et al.[80]) during DoS malicious attacks. The proposed

challenge-based trust model evaluated the satisfaction between desired replies and gained input to detect malicious nodes in HSN. The malicious IoMT nodes were placed in the same experiment environments along with the central server. When the trust values in HSN became stable, insider DoS attacks were launched. Repeat the test three times to limit the influence of unexpected circumstances. Figure 7 depicts the aggregate trust levels of malicious IoMT nodes. The following are the main points to consider.

1. According to most trust management methods, the trust values of malicious IoMT nodes started falling once the attack was launched with the forgetting factor that highlights the behavior of the IoMT node. The reputation of Fung et al. trust model drops quicker than Duma et al. trust model.

2. Duma et al. and Fung et al. both trust models reduce the reputation level slower than our proposed cluster-based hierarchical trust management scheme (Without a blockchain-based trust management approach). It happens because the hierarchical cluster's trust model calculates trust values based on packet status, which is quite sensitive to (malicious) traffic status. On the other hand, Fung et al.'s trust model experience a communication delay since it must first collect input from target nodes before evaluation.

3. When comparing our proposed blockchain-enabled trust model to the conventional hierarchical cluster-based trust management scheme, we discovered that the proposed method enhances the efficiency of IoMT malicious node detection in HSN contexts by one day earlier than its original (cluster-based hierarchical trust management scheme, Without blockchain). However, the proposed model faster degrades the trust value of malicious IoMT nodes below the threshold of 0.8. The key reason is that our technique uses the blockchain to allow HSN IoMT nodes to update their blocklist faster than the original system, and adjacent nodes interact with suspicious nodes to get additional traffic updates.

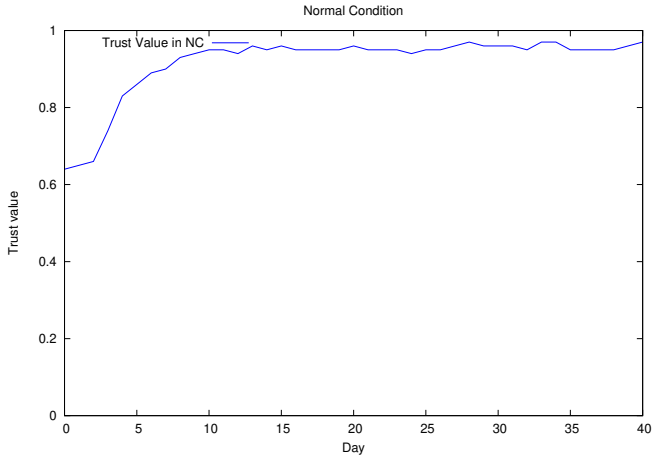

Figure 6. Average trust value in normal condition.

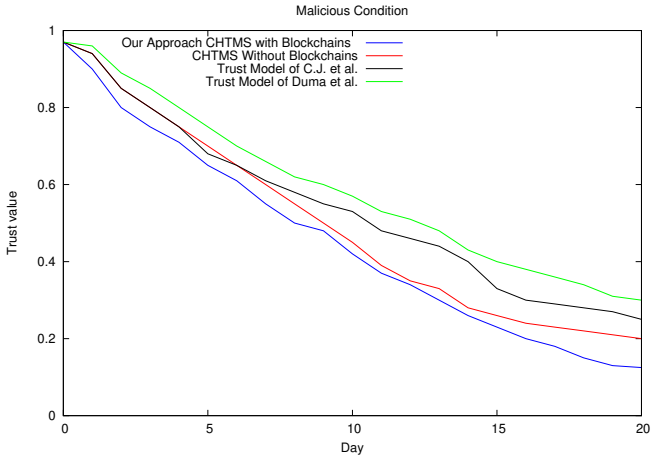

Figure 7. Average trust value in malicious condition.

Furthermore, our experiment results show that our proposed trust management scheme provides better malicious IoMT node detection performance than the original scheme and the other two comparable trust management schemes, based on the pattern of malicious nodes' reputation in the HSN environment. Therefore, the scalability of the proposed approach is considerably high in healthcare environments.

4.2 System Performance Analysis Undergoing DoS Attacks

The comparative analysis of Blockchain vs. Conventional MySQL database performance was recorded during HSN's simulated Denial of Service Attacks (DoS) attacks. During DoS attacks, the resource consumption is through sending queries to MySQL clients at port no 3306 and communicating with two Fabric blockchain nodes (mining node and the other is just a transaction receiving node). In HSN, simulate the malicious activity by frequently querying the Fabric nodes to get the most recent block information. On the other hand, querying MySQL for the total amount of entries in the table causes it to be delayed and yield invalid storage transactions. We experimented with two distinct levels of attacks: Low-Intensity attacks use ten concurrent processes, while High-Intensity attacks use twenty contemporary methods.

Both blockchain and MySQL nodes used the PTB-XL ECG dataset consisting of 21837 diagnostic 12-lead ECGs, each lasting 10 seconds, from 18885 patients. PTB-XL is a significant large electrocardiography dataset freely accessible on the Kaggle platform[82]. A vital diagnostic technique for determining a patient's heart state using IoMT-enabled electrocardiography (ECG). The Schiller AG devices used in collecting the waveform data that makes up the PTB-XL ECG dataset[83]. At the Physikalisch-Technische Bundesanstalt (PTB), the records were vetted and transformed into a systematically structured database over time. PTB-XL ECG dataset followed SCP-ECG standard while creating the ECG statements used to annotate the recordings.

MySQL takes advantage of the same data exchange module and execution platform that Blockchain nodes do. As a baseline, use MySQL database version 8.0 on Ubuntu 20.04.3 LTS for $x86\_64$ architecture, with an Intel Core 3.40 GHz i7-6700 CPU and 16 GB of RAM, as a comparison to our blockchain-enabled solution. After every time slot, all IoMTs submit the recorded data to MySQL, saving it if the request is authentic. We have constructed a table with two columns (ecgID and patientID), which is used to identify the record of patients and ECGs uniquely. We also query the complete list of records using MySQL's COUNT(*) function. The efficiency of this solution is measured using the two following metrics.

1. Transaction Failure Ratio (TFR): Every timeslot (ts), query the MySQL database and Hyperledger blockchain against the total number of currently processed transactions ΔT, and afterward compute the TFR ratio of unreachable data whose transaction queries could be delayed, restricted, or canceled. $TFR = 1 - \Delta T/S$, where $S = M * ts$, and $ts = 1, 2, ..., TS$.
2. Transaction Successful Ratio (TSR) is the ratio between the inserted data ΔI and the number of data storage requests found by counting the total number of transaction queries. TSR is computed when Hyperledger blockchain or MySQL database somehow does not process any queries, and IoMT nodes stop submitting requests $TSR = \Delta I/S$, in which $S = M * TS$.

Overall, the TFR and TSR ratios indicated the outcomes of the system operating under two malicious conditions (low-Intensity and high-intensity attacks). The TFR is computed immediately after processing each batch of ten transactions, while the system remains operational because it accepts and executes ongoing data requests. On the other hand, TSR is computed whenever all the system task is finished, i.e., when the IoMT nodes have performed the diagnostic process and the storage system has stopped handling requests. In some situations, if a transaction request fails while calculating TSR, it may be denied or canceled, so it is considered into TFR unless the reverse is true.TFR indicates the system's transient reaction characteristic, whereas TSR represents its robustness. Figure 8. demonstrated the impact of DoS attacks on the immediate transaction failure ratio, by modifying the number of IoMT nodes (two, three, and four) and the intensity of attacks (low and high). Our observations are given below:

1. When the count of storage transactions grows by IoMT nodes, some parts not be retrieved from the whole data, irrespective of its storage mechanism. As a result, some data storage queries are restricted during DoS attacks, regardless of whether the data is stored in a conventional database or a blockchain.
2. During DoS attacks, increasing the number of IoMT nodes and intensity level of attacks, the upper limit of TFR is 0.3 and 0.7 for conventional databases and blockchain, respectively. In this way, we can say blockchain has lower TFR than a conventional database, with the exact causes as discussed in the analysis of Dos attacks.
3. As demonstrated in Fig. 8., during low-intensity DoS attacks, the TFR is 0.1 in blockchain and 0.5 in database. While at the time of high-intensity DoS attacks, the TFR increases from 0.2 in blockchain and 0.63 in database due to the rise of blocked requests leading to TFR, along with the intensity of DoS attacks increasing.
4. The number of IoMT nodes increased from two to four at any moment, and the rate of TFR drop is high; with the low-intensity DoS attacks, it dropped from 0.2 to 0.05 and 0.6 to 0.45 blockchain and database, respectively. However, the same drop rate is also found in high-intensity DoS attacks. As discussed during the analysis of DoS attacks, more IoMT nodes yield high transaction requests, resulting in a lower TFR.

At last, Figure 9. demonstrated the performance of conventional databases and blockchains in terms of transaction success ratio undergoing DoS attacks. Our observations are given below:

1. The TSR of blockchain is not affected by DoS attacks; it always lies on the top. Conversely, the TSR of the database affected by DoS attacks improves whenever the number of transaction requests increases. The reason is the same as discussed in TFR analysis, the less amount of data exposed in DoS attacks increases the count of the transaction request in the IoMT multimode structure, which yields sluggish improvement in the TSR of the database. At the same time, the TSR of the blockchain remains unaffected.

2. When the number of IoMT nodes increases, the corresponding number of database storage requests also increases, so the TSR of the database improves. At the same time, the TSR of the blockchain remains unaffected at any intensity of DoS attacks.

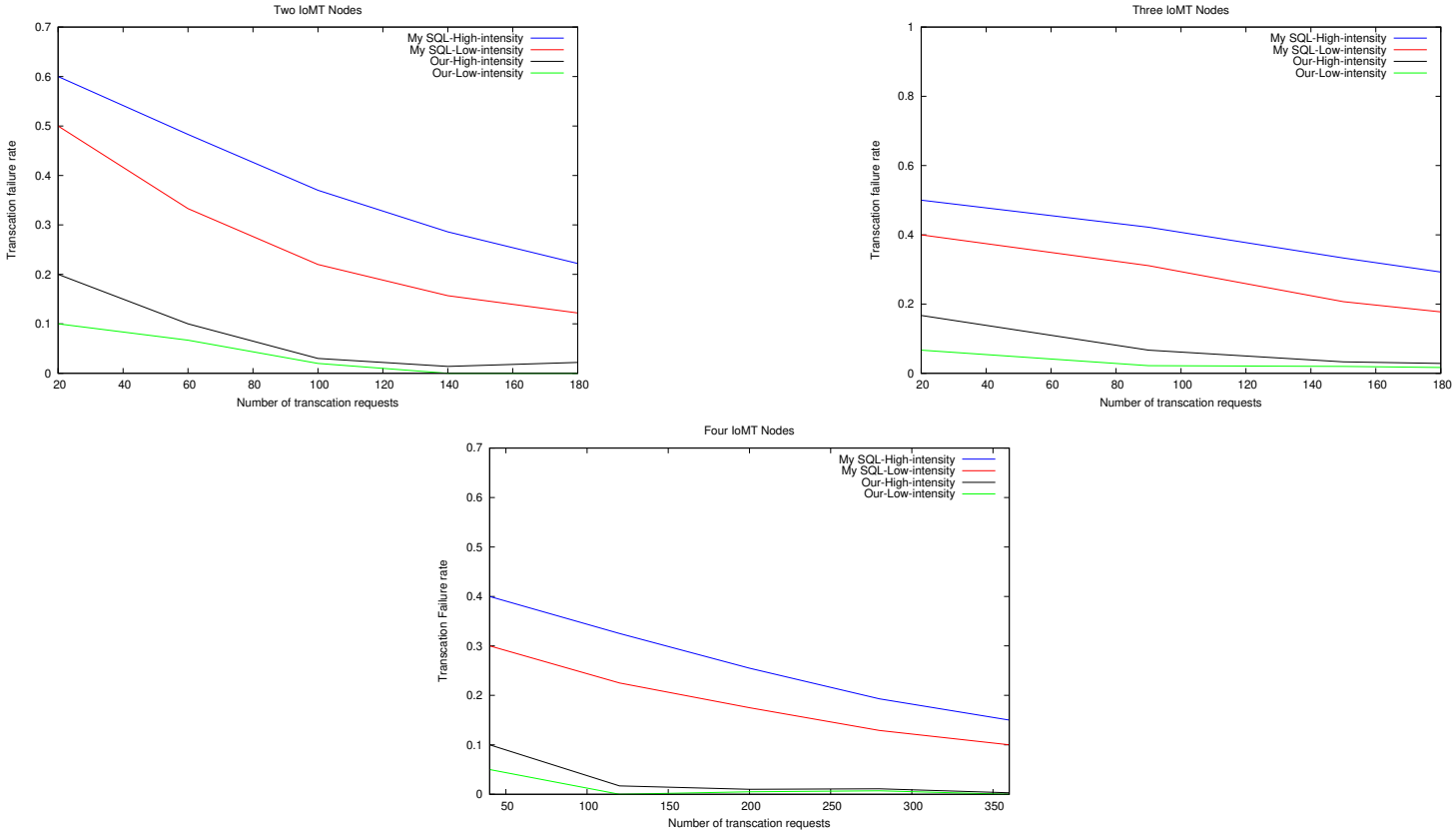

Figure 8. Two, three and four IoMT nodes undergoes DoS attacks impacts the immediate TFR.

4.3 Analyze Communication Cost in HSN

In the worst-case situation, each IoMT node wants to interact with all the IoMT nodes in their group. Each IoMT group wants to interact with all the groups in the HSN network. Suppose the HSN network comprises $|N|$ groups, and the average group size is η. When an IoMT node A intends to communicate with other IoMT nodes B via intragroup communication, then IoMT node A will request a maximum of $\eta-2$ peer recommendation. In response, an IoMT Node A get $\eta-2$ replies. The greatest communication overhead for each IoMT node A is $2(\eta-1)(\eta-2)$ to communicate with all IoMT nodes in the group. The highest intragroup communication cost (IC_{intra}) in proposed CHTMS is $2\eta(\eta-1)(\eta-2)$ if all IoMT nodes intend to communicate with every IoMT node in HSN.

When an IoMT group A intends to communicate with other groups B, it will submit the highest single peer recommendation request to the BS during intergroup communication. As a result, every request requires a two-packet communication cost. The highest communication cost can be $2|N|-1$ packets if group A tends to communicate with every group $|N|$ in HSN. The highest intergroup communication cost (IC_{inter}) of the CHTMS is $2|N|N-1|$ when every group tends to communicate with every group in HSN. As a result, the CHTMS's highest communication cost (CC) in the HSN is:

$$\begin{aligned} CC &= [No.\ of\ groups(N) * Intragroup\ communication\ cost(IC_{intra})] + intergroup\ communication\ cost(IC_{inter}) \\ &= [|N| * 2\eta(\eta-1)(\eta-2)] + 2|N||N-1| \\ &= 2|N|(\eta(\eta-1)(\eta-2)+(N-1)) \end{aligned} \tag{6}$$

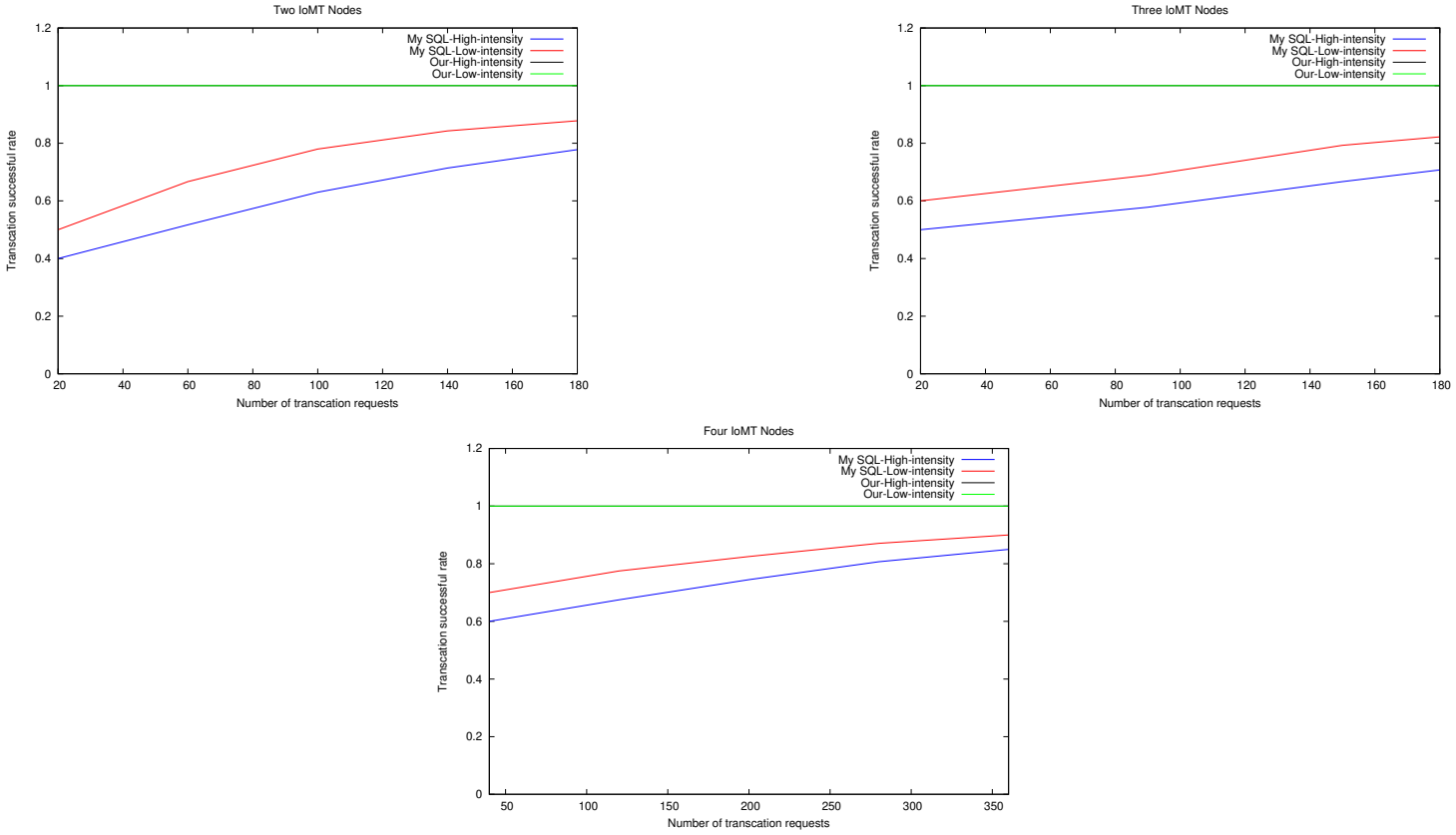

Figure 9. Two, three and four IoMT nodes undergoes DoS attacks impacts the immediate TSR.

4.3.1 Comparison of Communication Cost with other schemes

The communication cost of different trust management techniques for a large number of IoMT nodes ($N * \eta = 100\ nodes$) form N number of clusters with identical cluster sizes ($\eta = 10$) is shown in Table 3. Compared to the state-of-the-art approaches. The CHTMS levies less communication cost as even the number of clusters inside the HSN network grows. It also found that CHTMS is appropriate for large-scale HSNs with small cluster sizes (10 IoMT in each cluster). The crucial point concerning the ATRM scheme[84] is whether it displays the results of one transaction per node. For Instance, when IoMT node A and IoMT node B desire to communicate, they initially share four packets. The trust is computed again after finalizing the transaction when node A wishes to begin a new transaction with B. As a result, the ATRM scheme's communication cost will rise quarterly with each transaction. In the CHTM scheme, an IoMT node A wishes to initiate the new transaction with IoMT node B only when the first transaction is completed; no extra communication cost occurs since IoMT node A already computes the trust by considering the history of previous transactions(s).

Trust Management Scheme	Communication Cost
PLUS[85]	$2\|N\|(\eta(\eta-1)^2+(N-1)^2)$
ATRM[84]	$4\|N\|(\eta(\eta-1)+(N-1))$
RFSN[28]	$2\|N\|(\eta(\eta-1)(\eta-2)+(N-1)(N-2))$
CHTMS	$2\|N\|(\eta(\eta-1)(\eta-2)+(N-1))$

Table 3. Communication Cost of Trust Management Schemes in Worst Case.

4.4 Analyze CPU Utilization in HSN

Due to the advent of trust management procedures in HSN, it is realistic to expect additional computational effort on both the IoMT Node and Base-Station server sides.

1. Interactions with the HSN server, including monitoring of packet status, blocklist updating, and communication between IoMT nodes and Base-Station, are the significant causes of CPU load at the Base-Station server.

2. The leading causes of forging the CPU load at the smartphone and IoMT node side are interactions with the HSN IoMT node, including communication with HSN IoMT nodes, trust measurement, implementation of security policy, construction and updating blocklist, blockchain data retrieval.

In our experiments, Figure. 10 and 11 exhibits the CPU utilization at IoMT nodes and the Base-Station server in healthcare organizations under Normal and Malicious Condition, covering the highest, lowest, and average CPU utilization.

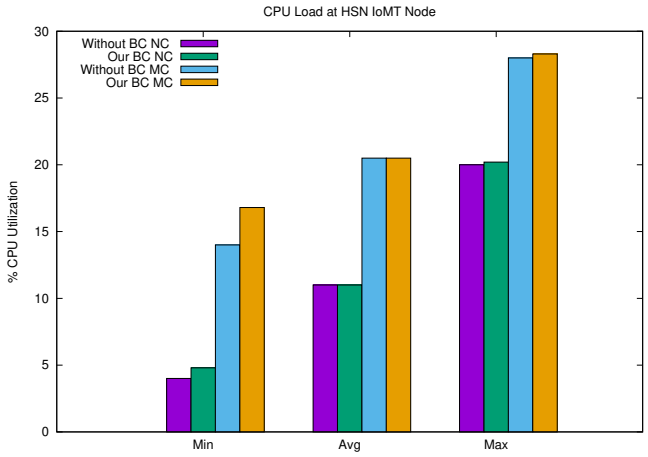

Figure 10. CPU Utilization at HSN IoMT Node.

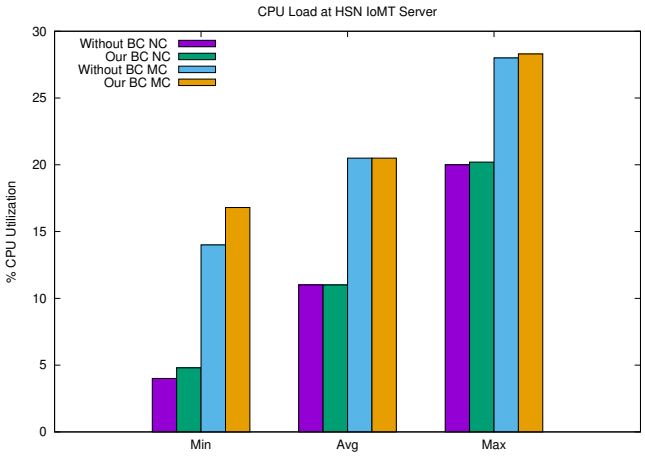

Figure 11. CPU Utilization at HSN Base-Station Server.

The following are the critical observations of the result analysis.

1. It is clear that the CPU demand was significantly higher under the Malicious Condition (MC) than in the normal state. As per observation, the average CPU utilization of the Base-station server is 20.3 percent in normal conditions, but its rises to 30.6 percent during malicious conditions. However, in this case, malicious traffic can generate more significant interaction between smartphones, HSN IoMT nodes, and the Base-station server.

2. The CPU load on the Base-Station server was significantly more than the smartphone and IoMT node side. The result represents 20.3 percent and 11 percent CPU loads for the Base-Station server and smartphone/IoMT nodes in HSN. The Base-Station server is responsible for various services, including data gathering, trust computation and assessment, blocklist production, and updating.

3. It has been found that our blockchain-based trust management technique boosts CPU utilization compared to the original trust management scheme without blockchain involvement. The average CPU utilization is 11 percent and 20.5 percent at the smartphone/IoMT node level, in normal conditions and malicious conditions, respectively, while 20.3 percent and 30.6 percent at the Base-Station server level, in normal conditions and malicious conditions, respectively.

4. The CPU workload between our blockchain-enabled technique and the original scheme (without blockchain) slightly increases initially, less than 1 percent in the case of average CPU utilization at both the IoMT node and BS-Server level. Therefore, we adapted the blockchain-enabled technique because it does not significantly increase CPU workload compared to many other security-related benefits over the original scheme.

5 Conclusion

The extensive usage of IoMTs in HSN has become a prominent target for attackers, who can compromise multiple healthcare resources inside the network. Moreover, insider attacks pose a significant danger to IoMT, necessitating effective trust management strategy implementation in healthcare. We proposed an algorithm for accurately collecting data from IoMT devices and exchanging patient data securely across IoMT nodes and smartphone devices. We integrate Hyperledger blockchain in HSN, backing a robust and lightweight CHTMS, which lowers the cost of trust analysis to find the level of trust and identify insider malicious /compromised IoMT nodes more proficiently and rapidly update the blocklist across HSN. Our proposed

CHTMS scheme provides better security, stability, and more resistance to malicious attacks such as denial of service (DoS), eclipse attacks, and terminal device failure. We find that the performance of the proposed approach is better in terms of TFR, TSR, CPU workload, and communication cost compared to the state-of-the-art approaches. Future research must improve blockchain-enabled trust management schemes by looking into latency issues and low processing power nodes like smartphones. Similarly, the researcher must develop distributed trust management approaches for healthcare organizations to make IoMT nodes more intelligent, ensuring data sharing is more secure and masking IoMT identification to ensure anonymity. So, the identity agnostic context is still challenging to establish and maintain trust among IoMT nodes.

6 Declarations

6.1 Ethical Approval

Not Applicable

6.2 Competing interests

The authors reported no potential competing interest.

6.3 Authors' contributions

(Author A[1]. Did experimental work, Wrote manuscript text, and prepared figures used in the manuscript and the rest of the authors B[2]. C[3].D[4]. are supervisors, so they carefully reviewed the manuscript.)

6.4 Funding

Not Applicable

6.5 Availability of data and materials

We used the PTB-XL ECG dataset publically online, accessible at the Kaggle platform[82].

References

1. Sentance, R. 7 examples of how the internet of things is facilitating healthcare (2021).
2. Lee, J.-H. Future of the smartphone for patients and healthcare providers. *Healthc. informatics research* **22**, 1–2 (2016).
3. Guo, J. Smartphone-powered electrochemical biosensing dongle for emerging medical iots application. *IEEE Transactions on Ind. Informatics* **14**, 2592–2597 (2017).
4. Yang, Z., Zhou, Q., Lei, L., Zheng, K. & Xiang, W. An iot-cloud based wearable ecg monitoring system for smart healthcare. *J. medical systems* **40**, 1–11 (2016).
5. Meng, W., Li, W., Xiang, Y. & Choo, K.-K. R. A bayesian inference-based detection mechanism to defend medical smartphone networks against insider attacks. *J. Netw. Comput. Appl.* **78**, 162–169 (2017).
6. The rise of the connected world - deloitte us.
7. Gartner survey reveals 47% of organizations will increase investments in iot despite the impact of covid-19.
8. Rathore, M. M., Ahmad, A., Paul, A., Wan, J. & Zhang, D. Real-time medical emergency response system: exploiting iot and big data for public health. *J. medical systems* **40**, 1–10 (2016).
9. Shelke, D. Y. & Sharma, A. Internet of medical things (iomt).
10. Symantec. networked medical devices: Security and privacy threats.
11. Alkinoon, M., Choi, S. J. & Mohaisen, D. Measuring healthcare data breaches. In *International Conference on Information Security Applications*, 265–277 (Springer, 2021).
12. atlanticcouncilFollow. The healthcare internet of things: Rewards and risks.
13. Hoffman, L. J., Lawson-Jenkins, K. & Blum, J. Trust beyond security: an expanded trust model. *Commun. ACM* **49**, 94–101 (2006).
14. Ng, H., Sim, M. & Tan, C. Security issues of wireless sensor networks in healthcare applications. *BT Technol. J.* **24**, 138–144 (2006).
15. Chen, J., Tian, Z., Cui, X., Yin, L. & Wang, X. Trust architecture and reputation evaluation for internet of things. *J. Ambient Intell. Humaniz. Comput.* **10**, 3099–3107 (2019).

16. Corradini, E., Nicolazzo, S., Nocera, A., Ursino, D. & Virgili, L. A two-tier blockchain framework to increase protection and autonomy of smart objects in the iot. *Comput. Commun.* **181**, 338–356 (2022).

17. Wei, L., Yang, Y., Wu, J., Long, C. & Li, B. Trust management for internet of things: A comprehensive study. *IEEE Internet Things J.* (2022).

18. Job, D. & Paul, V. Challenges, security mechanisms, and research areas in iot and iiot. In *Internet of Things and Its Applications*, 523–538 (Springer, 2022).

19. Heinzelman, W. B., Chandrakasan, A. P. & Balakrishnan, H. An application-specific protocol architecture for wireless microsensor networks. *IEEE Transactions on wireless communications* **1**, 660–670 (2002).

20. Younis, O. & Fahmy, S. Heed: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. *IEEE Transactions on mobile computing* **3**, 366–379 (2004).

21. Manjeshwar, A. & Agrawal, D. P. Teen: Arouting protocol for enhanced efficiency in wireless sensor networks. In *ipdps*, vol. 1, 189 (2001).

22. Lindsey, S. & Raghavendra, C. S. Pegasis: Power-efficient gathering in sensor information systems. In *Proceedings, IEEE aerospace conference*, vol. 3, 3–3 (IEEE, 2002).

23. Du, W., Deng, J., Han, Y. S. & Varshney, P. K. A key predistribution scheme for sensor networks using deployment knowledge. *IEEE Transactions on dependable secure computing* **3**, 62–77 (2006).

24. Lim, S., Pu, C., Chae, J., Min, M. & Liu, Y. Hide-and-detect: forwarding misbehaviors, attacks, and countermeasures in energy harvesting-motivated networks. *Energy Harvest. Wirel. Sens. Networks Internet Things* 271 (2022).

25. Ahmad, S. & Kim, D. A multi-device multi-tasks management and orchestration architecture for the design of enterprise iot applications. *Futur. Gener. Comput. Syst.* **106**, 482–500 (2020).

26. Xiong, L. & Liu, L. Peertrust: Supporting reputation-based trust for peer-to-peer electronic communities. *IEEE transactions on Knowl. Data Eng.* **16**, 843–857 (2004).

27. Saidi, A. Trust evaluation method for wireless sensor networks based on behavioral similarity and similarity coefficient. In *2021 International Conference on Networking and Advanced Systems (ICNAS)*, 1–6 (IEEE, 2021).

28. Ganeriwal, S., Balzano, L. K. & Srivastava, M. B. Reputation-based framework for high integrity sensor networks. *ACM Transactions on Sens. Networks (TOSN)* **4**, 1–37 (2008).

29. V, R. & M, S. Energy efficient hierarchical trust management scheme for solving cluster head compromising problem in wireless sensor networks. In *2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, 1–6, DOI: 10.1109/ICIIECS.2015.7192854 (2015).

30. Das, R., Dash, D. & Sarkar, M. K. Htms: fuzzy based hierarchical trust management scheme in wsn. *Wirel. Pers. Commun.* 1–34 (2020).

31. Chen, L., Lee, W.-K., Chang, C.-C., Choo, K.-K. R. & Zhang, N. Blockchain based searchable encryption for electronic health record sharing. *Futur. generation computer systems* **95**, 420–429 (2019).

32. Meng, W., Tischhauser, E. W., Wang, Q., Wang, Y. & Han, J. When intrusion detection meets blockchain technology: a review. *Ieee Access* **6**, 10179–10188 (2018).

33. Yu, E. Singapore suffers 'most serious' data breach, affecting 1.5m healthcare patients including prime minister (2018).

34. Radanliev, P. *et al.* Covid-19 what have we learned? the rise of social machines and connected devices in pandemic management following the concepts of predictive, preventive and personalized medicine. *EPMA journal* **11**, 311–332 (2020).

35. Sharma, A. *et al.* Blockchain technology and its applications to combat covid-19 pandemic. *Res. on Biomed. Eng.* 1–8 (2020).

36. O'Reilly-Shah, V. N. *et al.* The covid-19 pandemic highlights shortcomings in us health care informatics infrastructure: a call to action. *Anesth. analgesia* (2020).

37. Mandal, S. & Khan, D. A. A study of security threats in cloud: Passive impact of covid-19 pandemic. In *2020 International Conference on Smart Electronics and Communication (ICOSEC)*, 837–842 (IEEE, 2020).

38. Merazzo, K. J., Totoricaguena-Gorriño, J., Fernández-Martín, E., Del Campo, F. J. & Baldrich, E. Smartphone-enabled personalized diagnostics: Current status and future prospects. *Diagnostics* **11**, 1067 (2021).

39. Razdan, S. & Sharma, S. Internet of medical things (iomt): overview, emerging technologies, and case studies. *IETE Tech. Rev.* 1–14 (2021).

40. Wüst, K. & Gervais, A. Do you need a blockchain? In *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*, 45–54 (IEEE, 2018).

41. Zheng, Z., Xie, S. & Dai, H. X, chen, h. wang,"an overview of blockchain technology: Architecture, consensus, and future trends" in proc. In *IEEE International Congress on Big Data*, 557–564 (2017).

42. Javaid, M., Haleem, A., Singh, R. P., Khan, S. & Suman, R. Blockchain technology applications for industry 4.0: A literature-based review. *Blockchain: Res. Appl.* 100027 (2021).

43. Wood, G. *et al.* Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper* **151**, 1–32 (2014).

44. Sasson, E. B. *et al.* Zerocash: Decentralized anonymous payments from bitcoin. In *2014 IEEE Symposium on Security and Privacy*, 459–474 (IEEE, 2014).

45. Nakamoto, S. Bitcoin: A peer-to-peer electronic cash system. *Decentralized Bus. Rev.* 21260 (2008).

46. Ali, M., Nelson, J., Shea, R. & Freedman, M. J. Blockstack: A global naming and storage system secured by blockchains. In *2016 {USENIX} annual technical conference ({USENIX}{ATC} 16)*, 181–194 (2016).

47. Joshi, A. P., Han, M. & Wang, Y. A survey on security and privacy issues of blockchain technology. *Math. foundations computing* **1**, 121 (2018).

48. Brown, R. G., Carlyle, J., Grigg, I. & Hearn, M. Corda: an introduction. *R3 CEV, August* **1**, 15 (2016).

49. Cachin, C. *et al.* Architecture of the hyperledger blockchain fabric. In *Workshop on distributed cryptocurrencies and consensus ledgers*, vol. 310 (Chicago, IL, 2016).

50. Bhan, R. Hyperledger fabric for healthcare smartphone network. https://github.com/Ratibhan/fabric.git (2022).

51. Nimbalkar, P. & Kshirsagar, D. Analysis of rule-based classifiers for ids in iot. In *Data Science and Security*, 461–467 (Springer, 2021).

52. Satheesbabu, S., Gokulakrishnan, P. & Dhanalakshmi, N. The surveillance of intrusion detection systems and approaches. *INFORMATION TECHNOLOGY IN INDUSTRY* **9**, 1135–1150 (2021).

53. Meng, W., Li, W. & Zhou, J. Enhancing the security of blockchain-based software defined networking through trust-based traffic fusion and filtration. *Inf. Fusion* **70**, 60–71 (2021).

54. Bradbury, M., Jhumka, A. & Watson, T. Trust trackers for computation offloading in edge-based iot networks. In *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*, 1–10 (IEEE, 2021).

55. Li, W., Meng, W., Parra-Arnau, J. & Choo, K.-K. R. Enhancing challenge-based collaborative intrusion detection against insider attacks using spatial correlation. In *2021 IEEE Conference on Dependable and Secure Computing (DSC)*, 1–8 (IEEE, 2021).

56. Alevizos, L., Ta, V. T. & Hashem Eiza, M. Augmenting zero trust architecture to endpoints using blockchain: A state-of-the-art review. *Secur. Priv.* e191 (2021).

57. Li, W., Tug, S., Meng, W. & Wang, Y. Designing collaborative blockchained signature-based intrusion detection in iot environments. *Futur. Gener. Comput. Syst.* **96**, 481–489 (2019).

58. Gu, J. & Lu, S. An effective intrusion detection approach using svm with naïve bayes feature embedding. *Comput. & Secur.* **103**, 102158 (2021).

59. Karati, A., Islam, S. H. & Karuppiah, M. Provably secure and lightweight certificateless signature scheme for iiot environments. *IEEE Transactions on Ind. Informatics* **14**, 3701–3711 (2018).

60. Olariu, S., Xu, Q., Eltoweissy, M., Wadaa, A. & Zomaya, A. Y. Protecting the communication structure in sensor networks. *Int. J. Distributed Sens. Networks* **1**, 187–203 (2005).

61. Misra, S. & Xue, G. Efficient anonymity schemes for clustered wireless sensor networks. *Int. J. Sens. Networks* **1**, 50 (2006).

62. Ji, D. *et al.* Smartphone-based electrochemical system for biosensors and biodetection. In *Biomedical Engineering Technologies*, 493–514 (Springer, 2022).

63. Zhang, M., Cui, X. & Li, N. Smartphone-based mobile biosensors for the point-of-care testing of human metabolites. *Mater. Today Bio* 100254 (2022).

64. Prabhune, A. G. mhealth initiatives for non-communicable disease (ncd s)–a scoping review of the indian scenario. *resource* **1**, 0 (2022).

65. Sohrabi, H. *et al.* State-of-the-art cancer biomarker detection by portable (bio) sensing technology: A critical review. *Microchem. J.* 107248 (2022).

66. Bhan, R. 256-bit elliptic curve cryptography (ecc) parameters secp256k1 associated with a koblitz curve. https://github.com/Ratibhan/ECC.git (2022).

67. Vahdati, Z., Yasin, S., Ghasempour, A. & Salehi, M. Comparison of ecc and rsa algorithms in iot devices. *J. Theor. Appl. Inf. Technol.* **97** (2019).

68. Lie, X., Jiang, P., Chen, T., Xiapu, L. & Qiaoyan, W. A survey on the security of blockchain systems future generation computer systems (2017).

69. Heilman, E., Kendler, A., Zohar, A. & Goldberg, S. Eclipse attacks on bitcoin's peer-to-peer network. In *24th {USENIX} Security Symposium ({USENIX} Security 15)*, 129–144 (2015).

70. Bahga, A. & Madisetti, V. K. Blockchain platform for industrial internet of things. *J. Softw. Eng. Appl.* **9**, 533–546 (2016).

71. Liu, Z. *et al.* Combining graph neural networks with expert knowledge for smart contract vulnerability detection. *IEEE Transactions on Knowl. Data Eng.* (2021).

72. Marcus, Y., Heilman, E. & Goldberg, S. Low-resource eclipse attacks on ethereum's peer-to-peer network. *IACR Cryptol. ePrint Arch.* **2018**, 236 (2018).

73. Rajput, A. R., Li, Q. & Ahvanooey, M. T. A blockchain-based secret-data sharing framework for personal health records in emergency condition. *Healthcare* **9**, DOI: 10.3390/healthcare9020206 (2021).

74. Li, X., Jiang, P., Chen, T., Luo, X. & Wen, Q. A survey on the security of blockchain systems. *Futur. Gener. Comput. Syst.* **107**, 841–853 (2020).

75. Kouhizadeh, M., Saberi, S. & Sarkis, J. Blockchain technology and the sustainable supply chain: Theoretically exploring adoption barriers. *Int. J. Prod. Econ.* **231**, 107831 (2021).

76. Li, Y., Qiao, L. & Lv, Z. An optimized byzantine fault tolerance algorithm for consortium blockchain. *Peer-to-Peer Netw. Appl.* 1–14 (2021).

77. Xu, X. *et al.* Concurrent practical byzantine fault tolerance for integration of blockchain and supply chain. *ACM Transactions on Internet Technol. (TOIT)* **21**, 1–17 (2021).

78. Chen, G., Branch, J., Pflug, M., Zhu, L. & Szymanski, B. Sense: a wireless sensor network simulator. In *Advances in pervasive computing and networking*, 249–267 (Springer, 2005).

79. Buchegger, S. & Le Boudec, J.-Y. Self-policing mobile ad hoc networks by reputation systems. *IEEE communications Mag.* **43**, 101–107 (2005).

80. Grandison, T. & Sloman, M. A survey of trust in internet applications. *IEEE Commun. Surv. & Tutorials* **3**, 2–16 (2000).

81. Shaikh, R. A., Lee, S., Khan, M. A. & Song, Y. J. Lsec: Lightweight security protocol for distributed wireless sensor network. In *IFIP International Conference on Personal Wireless Communications*, 367–377 (Springer, 2006).

82. Ptb-xl ecg dataset.

83. Wagner, P. *et al.* Ptb-xl, a large publicly available electrocardiography dataset. *Sci. data* **7**, 1–15 (2020).

84. Boukerch, A., Xu, L. & El-Khatib, K. Trust-based security for wireless ad hoc and sensor networks. *Comput. Commun.* **30**, 2413–2427 (2007).

85. Yao, Z., Kim, D. & Doh, Y. Plus: Parameterized and localized trust management scheme for sensor networks security. In *2006 IEEE International Conference on Mobile Ad Hoc and Sensor Systems*, 437–446 (IEEE, 2006).