

## FedUni ResearchOnline

<https://researchonline.federation.edu.au>

Copyright Notice

This is the submitted version of the following article:

Uddin, A., Stranieri, A., Gondal, I., & Balasubramanian, V. (2020). Blockchain leveraged decentralized IoT eHealth framework. *Internet of Things*, 9 (March).  
Which has been published in final form at:

<https://doi.org/10.1016/j.iot.2020.100159>

Copyright © 2020 Elsevier Inc. All rights reserved.

# Blockchain Leveraged Decentralized IoT eHealth Framework

Md.Ashraf Uddin<sup>a</sup>, Andrew Stranieri<sup>a,\*</sup>, Iqbal Gondal<sup>a</sup> and Venki Balasubramanian<sup>a</sup>

<sup>a</sup>Internet Commerce Security Laboratory, Federation University Australia, VIC 3350, Australia

## ARTICLE INFO

### Keywords:

Internet of Things  
Blockchain  
consensus mechanism  
Proof of Stake  
Patient Agent  
Fog Computing  
Edge Computing  
Cloud  
Patient Monitoring  
eHealth  
Fuzzy Inference Process  
Task Offloading.

## ABSTRACT

Blockchain technologies recently emerging for eHealth, can facilitate a secure, decentralized and patient-driven, record management system. However, Blockchain technologies cannot accommodate the storage of data generated from IoT devices in remote patient management (RPM) applications as this application requires a fast consensus mechanism, careful management of keys and enhanced protocols for privacy. In this paper, we propose a Blockchain leveraged decentralized eHealth architecture which comprises three layers: 1) The Sensing layer- Body Area Sensor Networks include medical sensors typically on or in a patient body transmitting data to a smartphone. 2) The NEAR processing layer- Edge Networks consist of devices at one hop from the data sensing IoT devices. 3) The FAR processing layer-Core Networks comprise Cloud or other high computing servers). A Patient Agent (PA) software replicated on the three layers processes medical data to ensure reliable, secure and private communication. The PA executes a lightweight Blockchain consensus mechanism and utilizes a Blockchain leveraged task-offloading algorithm to ensure patient's privacy while outsourcing tasks. Performance analysis of the decentralized eHealth architecture has been conducted to demonstrate the feasibility of the system in the processing and storage of health data.

## 1. Introduction

Recently a wide range of IoT health applications has been developed to automate and make health services accessible to individuals. For instance, remote patient management (RPM) covers a variety of health services, including continuous vital signs monitoring with wearable or implantable sensors, arrhythmia detection, fall detection, oxygen therapy regulation, monitoring of pregnant women, chemotherapy reactions, and glucose monitoring[1].

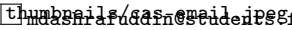
However, eHealth services have not flourished to the extent expected due in part to challenges associated with reliability, fault tolerance, and privacy challenges. In eHealth, patient's physiological data, captured by medical IoT(Internet of Things) devices is transmitted to Edge or Cloud entities managed by third parties, which makes data security and the preservation of a patient's privacy challenging.

Most IoT systems are centralized in that data flows to a single server for processing and storage. This makes these systems vulnerable to a single point of failure, particularly while handling a large number of end-to-end communications [2]. Cyberattacks such as Ransomware, and Denial of Service(DoS) attacks can paralyze conventional eHealth systems and dangerously disrupt healthcare services [3]. Recently, health data has become more attractive to attackers; the US Department of Health and Human Services(HHS) reported over 2250 data breaches between 2009 and 2018[4]. Further, insiders such as healthcare professionals, support staff, and service providers are associated with over half of recent health data breaches[4].

A typical IoT architecture involves a single instance of a health app maintained on a smartphone transmitting data to Edge devices then Cloud servers which are managed by third parties. This architecture is vulnerable to eavesdropping attacks over Bluetooth, Zigbee, or WiFi links, a man in the middle attack, DoS, and insider attacks. Further, as outlined, conventional Edge[5] or Cloud process[6] cannot guarantee accountability and tractability of patient's data due to third party involvement.

The processing of data in the Cloud in the typical IoT architecture requires a high level of accountability[7] and transparency over how data in the Cloud servers are used. Statutes and regulations for the regulation of health data now exist; however, Cloud service providers compliance with them is difficult to ascertain. This contributes to undermining the trustworthiness of Cloud-based processing of data.

The mobile to Cloud component of the IoT architecture called Mobile Cloud Computing(MCC) [8][9] has extended the capabilities of a smartphone by enabling uploading health data to Cloud server for processing and storage. However,

 [mdashrafuddin@federation.edu.au](mailto:mdashrafuddin@federation.edu.au) (Md.Ashraf Uddin); [a.stranieri@federation.edu.au](mailto:a.stranieri@federation.edu.au) (A. Stranieri); [iqbal.gondal@federation.edu.au](mailto:iqbal.gondal@federation.edu.au) (I. Gondal); [v.balasubramanian@federation.edu.au](mailto:v.balasubramanian@federation.edu.au) (V. Balasubramanian)  
ORCID(s):

the efficacy of MCC depends on the extent to which connectivity loss and latency can be minimized. Without this, the large storage and processing capacity of Cloud environments, cannot be utilized because of excessive transmission delays and unstable connections.

Multi-access Edge Computing (MEC)[10], also known as Mobile Edge Computing, provides an IT service environment at the edge of the cellular network and closer to the customer to access Cloud computing capabilities. MEC supports distributed edge computing by processing content on Edge devices such as base stations, radio network controllers, hot spots, local data centres, routers, switches, and WiFi access points. Collecting and processing data closer to the customer reduces latency and brings real-time performance to applications requiring high-bandwidth. Most Edge computing initiatives are being developed using open-source hardware and software that leverage Cloud and virtualization paradigms, including SDN(Software Defined Network) and NFV(Network Function Virtualization). The MEC platform supports multi-tenancy for cellular operators to rent their radio access network to authorized third parties such as application developers and contents providers. In distributed Edge network of MEC, an Edge server with lower processing capabilities can outsource its tasks to remote Edge servers with higher computing capabilities. But, migrating tasks to remote Edge servers introduce security threats that can result in data theft, and privacy breaches because diverse stakeholders manage Edge computing devices.

In this article, a Blockchain is deployed to perform task migration on the Edge network. A Blockchain consensus mechanism is executed on the MEC to provide its tenants and customers with faster processing and higher security. This adaptation to IoT architectures and MEC is very important for dealing with health data where privacy concerns, the need to trust providers and quality of service demands are very high. This is particularly true for patient-generated health data.

Patient-Generated Health Data(PGHD)[3] such as biometric data, symptoms, lifestyle choice, and treatment history collected through sensors, mobile apps, web protocols, and home monitoring devices are distinct from healthcare data within a clinical setting. Ideally, a patient has control over how and where their patient-generated data is stored and shared. Patients may need to share their generated data with diverse health care providers who each, typically maintains their own electronic medical record[11]. Patients may also choose to include their generated data to electronic health record(EHR)[12], an inclusive record of patient medical history, more extensive than medical records, to be shared and accessed by authorized users from across different healthcare providers.

An eHealth system is required to process, cleanse, analyze, and manage the patient-generated data to ensure it is accurate, complete, accessible to authorized users, and understandable to healthcare providers and meets patient's Quality of Services(QoS) expectations. These tasks require considerable computational resources, particularly cannot be achieved without compromising performance or security aspects of QoS expectations. Recently researchers have applied Blockchain technology to address privacy, security and third parties trust concerns regarding the management of medical data and healthcare services. Blockchain[13],[14] can support access control, secure storage and sharing of medical data without the need to trust third parties or intermediaries while maintaining user's privacy. Blockchain-based ehealth architectures [1, 15, 16] have been designed to manage health data autonomously. Further, distributed ledger technology, popularly known as Blockchain, has also been investigated to manage user's identity[17], metadata[18], share, and maintain logs of medical records[19] and patient key management[20][21]. For instance, MedRec[19], a prototype of a decentralized Blockchain architecture, was first implemented to contribute to an interoperable EHR system. In this eHealth system, Ethereum smart contracts orchestrate contents across different storage and provider sites. The system facilitates a comprehensive medical record view, care auditability, and data sharing through governing authentication logs. Linn[18] suggested an off-chain to manage raw health data and Blockchain to store metadata of medical records to tackle the challenges of accommodating decentralized ledger technology in healthcare.

However, the adoption of Blockchain in healthcare has mixed reviews from the researchers. Inclusion of Blockchain technology in healthcare is a trade-off between security and computational cost and storage resources.

Execution of Blockchain's algorithms, including mining, authorizing creates a burden on resource-constrained devices such as medical sensors and other IoT devices. The underlying core component of a Blockchain, the consensus mechanism refers to a common agreement amongst Blockchain Miners about the Block's state. Consensus mechanisms applied in various Blockchain-based eHealth research require very high computational resources. For instance, the Proof of Work consensus mechanism applied in [15, 16, 22] needs high computational overhead, long delays, and a great deal of power. Dwivedi [1] attempted to reduce the computational overhead on IoT devices by deploying a Gateway node to gather data Blocks from a group of IoT devices. The Gateway verifies the Blocks as a Miner before adding the Blocks to the Blockchain overlay network. Tuli et al. [22] presented a generic Broker in the Fog to adopt

Blockchain into the Internet of Things data streams. The broker assigns Blockchain tasks to various Fog devices so that the computational challenges can be met. However, these approaches are still vulnerable to DoS attacks and tampering because they still rely on a centralized Blockchain controller. Fault tolerance capabilities of such eHealth architecture have not been significantly improved. Fault tolerance[23] refers to the capability of a system to continue its operation or keep delivering uninterrupted services even if its significant components or partial components stop working. A system can be made fault-tolerant by adding a software module that continuously monitors over the activities of functional units or replicating the system among multiple hardware.

The approach advanced here involves the creation of multiple instances of a Patient Agent which is a software module dedicated for a patient. The multiple instances of the Patient Agent reside in three levels; patient's smartphone, Fog devices and Cloud servers.

However, the advancement of digital health soars the demands for various medical data services such as advanced user-centric applications at an affordable price, enabling context-aware and proximity services, service delivery crowded areas, and advanced multimedia centric services. The vision of future 5G system, which is going to be evolved in its full-pledged form by 2020, can provide the customer with the above outlined customized and advanced health-care services. 5G network has promised to provide faster speeds and more reliable communication connections on smartphones and other devices. 5G network is defined more beyond offering low latency and ultra-high bandwidth to services. 5G has brought the concept of network slicing into reality with the help of Software Defined Network(SDN) and Network Function Virtualization(NFV) technologies. Network slicing means to create multiple isolated end-to-end virtual/logical networks on top of a shared physical network[24] to meet the resource requirements for diverse kinds of applications. Each logical network possesses dedicated and shared resources in terms of processing power, storage, traffic capacity, connectivity, and coverage latency, and adequate bandwidth required for an application[25]. 5G technology is envisioned to support a wide range of applications, including IoT, vehicular network [26], and healthcare[27]. Healthcare has a diverse range of applications such as mobile health, assisted driving, aged care. These applications require various levels of resource requirements(processing power, storage, and bandwidth). For instance, an emotion-aware application supporting real-time emotion detection should operate on a network slice, offering low latency to ensure the patient's QoE(quality of experience). In remote surgery applications, a network slice with ultra-low latency and highly reliable connections are required to ensure the safety and security of the patient. In this article, we describe how the Patient Agent and the Blockchain can be envisaged on the 5G network to manage and allocate logical resources to diverse health applications to improve Quality of Experience(QoE).

In this paper, we focus on developing a Blockchain leveraged eHealth framework. The main contributions of this paper are summarized as follows:

1. An eHealth system leverages Blockchain technology with an architecture consisting of three layers; the sensing layer(Body Area Sensor Network), NEAR processing layer( the Fog) and FAR processing layer( the Cloud). Multiple instances of the Patient Agent named smartphone Agent, Fog Agent and Cloud Agent are deployed at three layers to process patient's data.
2. A customized Blockchain with a lightweight modified Proof of Stake consensus mechanism implemented at the NEAR processing layer to process data streamed from medical IoT sensors. The consensus mechanism for the healthcare Blockchain is executed on the Edge devices to best use those device's faster communication capacity leaving the permanent storage for the Blockchain to be managed in the Cloud.
3. Proposal of a Blockchain leveraged Task Migration Algorithm. The method migrates tasks to neighbouring or remote Fog Agents based on data sensitivity where remote Agent's profile is managed on the Blockchain.

The rest of the article is structured as follows. We review related research in Section 2 and describe our proposed eHealth system in Section 3. The performance of the proposed approach is presented in Section 4. The security analysis of the architecture has been performed in Section 5 before concluding the article in Section 6.

## 2. Literature Review

In this section, we review literature in four categories: conventional healthcare architecture, Blockchain-based healthcare architecture, 5G enabled eHealth and Blockchain consensus mechanism.

### 2.1. Conventional Fog/Cloud Healthcare Architecture

Recent eHealth architectures [28], [29],[8] incorporated Fog computing with smartphone and Cloud to make the processing of health data faster. Mahmud et al. [28] presented an IoT eHealth system where pre-processing including

data compression, filtration and analytics are performed on the Fog devices before transmitting data to Cloud servers for advanced processing. Rahman et al. [29] advanced an eHealth prototype by exploiting a smart gateway at Edge network. The smart gateway implemented using UT-Gate supports a good range of high-level services, including local storage, real-time local data processing, early warning, and embedded data mining. The health data processed by the gateway is transmitted to the Cloud servers for future access.

Verma and Sandeep[30] suggested an eHealth architecture with a feature of diagnosis. A gateway or local processing unit residing at Fog layer receives health data from medical IoT devices to perform pre-processing. They advanced a disease diagnosis module in the Cloud subsystem. The module generates alerts and warnings for caregiver subsystems. Fernandes et al. [31] designed a multi-agent-based remote patient monitoring system where the roles of the agents cover a wide range of activities including identification, collection, storage, recovery, visualization, monitoring anomalies, resource notification and dynamic reconfiguration. The authors also constructed IoT4Health as an eHealth solution. Jin et al. [32]'s eHealth system integrated IoT devices and Cloud services. The IoT devices transmit health data to a mobile access point(MCP) to upload the data in the Cloud. The MCP chooses a Cloud service provider while optimizing bandwidth and maintaining the service deadline. Aazam et al. [8]'s eHealth system included a resource provisioning technique in the Fog. The technique used the relinquish probability model to reduce the wastage of resources of edge layer devices.

## 2.2. Blockchain Based Healthcare Architecture

Although the aforementioned eHealth frameworks have explored Fog/Edge computing for rapid access and processing of medical data, patient's security and privacy are not addressed in those proposals. Both Fog and Cloud involve their administrators to process medical data which threatens patient's privacy. Blockchain implemented over Fog and Cloud might enable the processing and storage of patient data while avoiding the reliance on the centralized Fog/Cloud administration. Blockchain structure has motivated researchers to devise privacy-preserving eHealth systems.

Uddin et al.[15] proposed a Patient-Centric Agent residing in the Smart Gateway to determine storage, access control and privacy level during the insertion of patient abnormal medical data into a customized Blockchain. The Patient-Centric Agent also selects Blockchain providers to schedule medical data for processing and storage. In [16], the role of the Patient Agent is extended to manage multiple Blockchains and multiple storage mediums, including Local Computer, Cloud to preserve patient's privacy. Tuli et al.[22] presented FogBus that is a lightweight Blockchain-based Fog computing framework. They introduced a universal broker software executing on the Fog device to merge Blockchain with Edge devices such as medical sensors. The broker schedules jobs among other devices in the Fog. However, a universal broker system in eHealth causes security and privacy threat for the patients. Rahman[33] proposed a secure therapy framework, including Blockchain at MEC(Mobile Edge Computing) and the Cloud. The therapy data from physician and patients are processed by Cloud and MEC Blockchain nodes to ensure immutable, anonymous, secure and transparent sharing. The Blockchain stores only hashes of the therapy multimedia and the actual multimedia data containing images, audios, videos are stored off-chain in a separate database. Although the framework includes MEC Blockchain to avoid shortcomings of high bandwidth and analytical processing required by the Cloud, the Ethereum consensus consumes high power at MEC.

Griggs[34] presented an architecture for automated remote patient monitoring using smart contract of the Ethereum. A smart device such as mobile or laptop collects and aggregates data transferred by body area sensors. The smart device sends the aggregated data to pre-specified smart contract stored on the Ethereum. The smart contract processes the data and sends the result and notification to smart devices and healthcare providers. The Blockchain only keeps a record of the event's occurrence, and data is stored on the Electronic Health Record(EHR). However, the smart devices can cause a single point of failure and be vulnerable to Denial of Service attack. The architecture only ensures the secure processing of medical data. Chen et al.[35] discussed a Blockchain-based medical data access framework, including the Cloud to store medical data. All kinds of communications between a patient and the third party insurance or healthcare professionals are logged in the Blockchain. Liang[36] developed a Blockchain integrated personal health data sharing framework where a mobile app gathers data from wearable sensors and inserts data into the Cloud and Blockchain hyperledger to verify the integrity of the data. Zhang[6] applied Blockchain-based architecture called FHIRChain to securely and scalable share clinical data. The requirements defined by "Shared Nationwide Interoperability Roadmap" were focused in the architecture. Brogan[37] described the role of distributed ledger technologies to advance the electronic health record, ensuring the authenticity and integrity of health data. The authors also demonstrated the application of IoT protocol masked authentication messaging extension module for securely sharing, storing and



retrieving encrypted data using a tamper-proof distributed ledger. Gordon[38] discussed the facilitation of Blockchain in patient-driven or patient mediating data interoperability with respect to health data accessibilities, aggregation, liquidity, identity and immutability. Rupasinghe[39] identified two categories of risk factors of fall, namely medical factors and environmental factors. All identified risk factors are labelled as weak, moderate and strong based on evidence and expert opinions. Four types of users put fall-related data to a consortium Blockchain to ensure the interoperability, accessibility, and availability of the data to predict the likelihood of fall of the aged people. The smart contract is proposed to perform user's registration, insertion of data into Blockchain and fall prediction.

Dwivedi et al.[1] presented a Blockchain oriented eHealth frameworks inspired by Ali et al.'s proposal of a lightweight Blockchain for IoT [40]. They considered an overlay network which is a peer-to-peer computer network built on top of another network where nodes are logically or virtually connected. In the proposal, the IoT medical devices generate Block to be verified by the cluster head of the overlay network before sending them to the Cloud servers. The authors utilized several lightweight standard security protocols to enforce security and preserve patient's privacy in the eHealth system. A static cluster head always verifies the integrity of the Blockchain. However, avoidance of a global consensus mechanism can weaken the sustainability of Blockchain-based eHealth system. We advanced our eHealth by running a lightweight consensus mechanism on a peer-to-peer Fog network. The cluster head nominated for a certain period based on nodes' properties execute the consensus mechanism. Further, the functional blocks required for monitoring a patient is bundled in the form of a Patient Agent (PA), and the PA is replicated in devices at three levels: smartphone, Fog and Cloud levels.

Gaetani[41] proposed a Blockchain comprised of two layers in the Cloud computing environment. The Blockchain in the first layer keeps a record of operations issued on the distributed database while avoiding the computationally expensive Proof of Work. The Blockchain in the second layer records the logged operations generated from the first layer database using Proof of Work. Novo[42] designed a Blockchain-based decentralized architecture to manage access control for memory and power-constrained IoT devices. The key feature of the architecture includes a manager hub between IoT devices and Blockchain. Manager hub requests Blockchain node for access policies stored in the Blockchain for a particular wireless sensor network. The smart contract is executed to insert access policy into the Blockchain.

Existing Blockchain eHealth architectures reviewed above included Fog and Cloud for the storage and processing of patient's data. However, those studies related to healthcare did not advanced the notion of executing Blockchain's controller in a distributed fashion at multiple levels like sensing, near and far processing levels. Decentralized Blockchain controller makes an eHealth system fault-tolerant, reliable and protects it from the DoS. Further, Blockchain based existing eHealth architectures did not advance consensus mechanism and privacy aware task offloading method. To bridge this research gap, we proposed a decentralized Patient Agent-based eHealth architecture with Blockchain implemented at Fog and Cloud level described in the next section. Here, a comparative analysis of Blockchain-based health records and conventional health records systems is presented in Table 1.

### 2.3. State-of-the-art 5G enabled eHealth systems

5G enabled some eHealth systems have recently been developed to support various personalized human-centric interactive applications. Chen[50] proposed a MEC (Mobile Edge Cloud) based emotion-aware architecture using the features of 5G networks such as high data rate, low latency and high computing capacity. In this proposal, mobile devices collect emotion-related information and send information to Cloudlet and remote Cloud for processing. LIN et al. [51] designed a system to handle big data in emotion-aware application using SDN and 5G technology. They described the functionalities of data collection, transmission and storage for the emotion-aware application. The wearable sensors and other devices without having sensor sense of patient data. Data is transmitted to the control layer of SDN with high throughput capacities and then forwarded to the data centre via SDN application. Finally, data is also uploaded in the Cloud for analyzing and storage. Hossain et al.[52] also presented a 5G enabled framework to recognize and monitor emotion using speech and image. Authors added a component to recognize emotion in a 5G based cognitive healthcare framework. They extracted features from the captured image using local binary pattern (LBP) and interlaced derivative pattern (IDP). The structure incorporated Bluetooth technology so that caregiver can estimate the precise location of the client.

Chen[27] presented a healthcare architecture focusing on data-driven computing and caching in 5G networks. Chen included an SDN based resource cognitive engine that optimally allocates resources analyzing patient's need in diverse situations. Further, they introduced caching called small cell Cloud existing in the mobile and Fog devices and macrocell residing in the Cloud to provide the user with better QoE(Quality of Experience) in a 5G network. Chen[53]

**Table 1**

The comparative analysis of conventional healthcare system and Blockchain based healthcare system

Parameter	Conventional Healthcare	Blockchain Healthcare
PIA(Privacy, Integrity, Availability)	Created and managed by healthcare professionals or national government or patients themselves where record managers can access a patient's record without the patient's knowledge. Health data is stored in off-premise third party providing servers such as Cloud server. Consequently, data integrity cannot be guaranteed[43] in these systems. Operational failures can make health services unavailable.	Patient-driven health record management system. Privacy is at risk if an attacker can correlate record transactions to the patient by analyzing the transaction's contents[44]. Blockchain Consensus mechanism ensures data integrity. Replication of complete health record amongst multiple entities can guarantee uninterrupted health services.
Freshness	Attackers might manipulate local timestamp of a centralized server.	Timestamped Block is added to the Blockchain after Miner's verification and the attacker cannot alter a Block's timestamp
Cyber Attack	Dissemination of patient records has real-world consequences, including diverse cyber-attacks: DoS, ransomware. Various restrictions are imposed on sharing massive health data in conventional EHR due to the risks of leakage of personal information[45]	The Blockchain healthcare has been proven to safeguard the system from DoS, ransomware but also suffers from some other attacks depending on the Blockchain algorithm such as long-range attack, mining and storage attack and known 51% attack.
CAP(Cost, Access, Performance)	Involvement of high deployment, maintenance and administrative costs[46]. Access is delayed due to fragmented health record among diverse health record systems	Government or stakeholder does not require a massive amount of deployment cost since individual entity contributes resources. Instead, cost-saving and profitable because a user is rewarded for taking part in mining. This can alleviate employee wages, legal cost, and data centre rentals [47] but outputs low throughput and high power consumption depending on the consensus mechanism.
SI(Standardization, Interoperability)	Maintenance of standardization across the various agencies[48]. Diverse health institutions or providers may have various legal requirements which add an extra barrier to the cross-border sharing health data. Different security and privacy methods used in such systems raise the interoperability issue among them.	These systems are lacking high-level standardization which obstructs the fast development of decentralized ledger technology in health sector[43]. But, universal set of rules and regulations of such technology can offer a high level of interoperability to share records across diverse institutional healthcare professionals.
AT(Auditability, Trust)	Logging information for every access, but auditing is not transparent due to the involvement of a single controller or institution. Resilience and sustainability of such a system rely mostly on single third party	Data is recorded in an immutable distributed ledger with auditing traces which guarantees transparency in the procession of data exchange [45]. Every entity maintains logging. Transparent sharing of health data by running a consensus mechanism, which substitutes the third party trust
Tamper-proof storage	Public or private Cloud generally handles the storage and processing of Big health data [49]. These repositories are unable to guarantee immutability of health record	Digital signature, the cryptographic linkage between Blocks, and a consensus mechanism make reliable and fault-tolerant storage. The Blockchain ledger replicated among multiple servers prevent a record from being tampered.[49].

also proposed a 5G enabled smart diabetes system with analysis of diabetes patient's suffering using a machine learning method. A social networking based data sharing model is also presented for 5G smart diabetes.

Sharma[54] presented a Blockchain oriented distributed SDN controller architecture for IoT networks. Traditional

SDN controller is centralized. Sharma proposed to use distributed SDN controller that would be connected using Blockchain technology. This architecture can withstand major cyber-attacks, whereas centralized SDN controller is vulnerable to a single point of failure. However, the incorporation of Blockchain at SDN controller level might have delay and additional computational cost in processing of the user's data using API(Application Programming Interface).

State-of-the-art research focused on the design of 5G enabled distinct health applications. However, a patient might need services from more than one health applications at a time. For instance, a patient might require both 5G enabled emotion detection and diabetic monitoring applications at the same time. Therefore, there needs a dedicated module to manage and control health applications personalized for a particular patient and determine resource requirements for those patient's health applications.

## 2.4. Consensus Mechanism in Blockchain

Gai[55] presented a Proof of Reputation based consensus mechanism for a peer-to-peer Blockchain network where a service provider receives some reputation from service requestors as incentives. A group of Miner is selected based on their earned reputation. The publications of their reputation transactions are blocked by the system instead of digital coin if a Miner's malicious activities are detected by the system. However, only reputation revoking cannot prevent Miner from performing malicious activities on the Blockchain. Further, some nodes can collude to work for the requestors and obtain high reputation.

Li[56] proposed a Proof of Vote(PoV) consensus mechanism where a group of commissioners vote for the selection of butler who are responsible for mining on the Blockchain. PoV outputs higher performance in terms of power and delay. But PoV is not fully decentralized like Proof of Work rather it provides controllable security, convergence reliability, and only one Block is confirmed within a time frame.

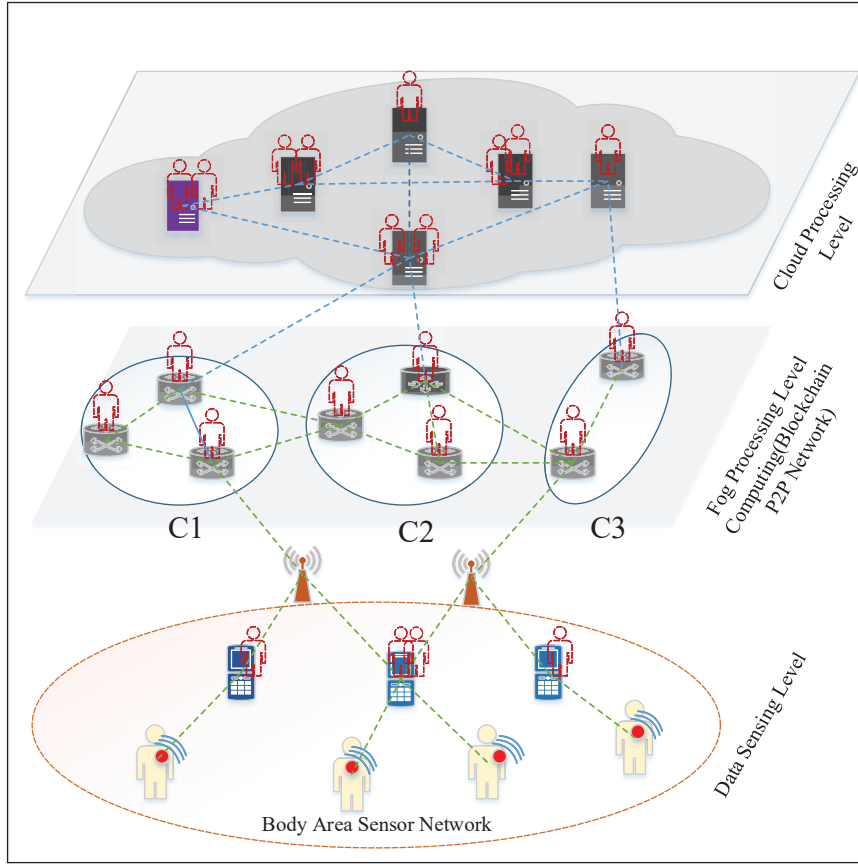
Bitcoin-NG proposed by Eyal et al. [57] selects a Miner through broadcasting macro Block on the Blockchain. The node that comes up with target hash(Proof of Work) for the micro Block mines the data Block. The Bitcoin-NG can reduce the latency and power consumption required for broadcasting data transaction throughout the network. Peterson et al.[58] followed MultiChain[59]'s PoW to randomly select Miner to perform validation on the Blockchain. Random Miner selection improves the throughput and reduce computational overhead. However, malicious and inefficient Miner might be nominated as Miner in random selection process. A hybrid consensus mechanism including reputation, voting, performance, randomness and stake of coin can meet challenges related to consensus protocol.

## 3. Decentralized Patient Agent based eHealth Framework

In this section, our eHealth framework is described in details. Figure 1 depicts the high-level view of the eHealth architecture that comprises devices from three levels: sensing, NEAR processing and FAR processing level. Devices at three levels contain multiple instances of a Patient Agent. The Patient Agents at the upper two levels: NEAR and FAR processing layers collaboratively take part in realising Blockchain technologies to process patient's data securely. An Agent dedicated to providing a patient with health services has several functionalities, including handling task migration, managing Blockchain, and network slices. The Agent's **Migration Handler(MH)** decides to migrate a task or locally execute the task using **Profile Monitoring(PM)** that collects profile information of remote Agents from the Blockchain. The **Execution Unit(EU)** locally processes a patient's health data. The **Blockchain Manager(BM)** of the PA assists EU, MH and Storage Management(SM) to accomplish their activities through a consortium Blockchain(Consortium Blockchain(CB) is not granted to a single entity like private Blockchain. Instead, CB, a semi-decentralized system, is managed and controlled by a group of approved entities.). The details about PA's functionalities depicted in Figure 4 are discussed in the later section. The three layers of the eHealth system are discussed below.

- **The Sensing Layer:** Wearable sensors, implantable sensors, smartwatch, smartphone and other IoT devices sense patient's vital signs and passive data, including room temperature, humidity. The IoT devices at the sensing layer are wirelessly connected to a smartphone via star topology. These devices transmit a patient's physiological sign, including ECG, EEG, BSC to the smartphone using Bluetooth or ZigBee protocol[60]. The smartphone instance of a PA performs pre-processing such as filtering noise, classification on physiological data to send them to the next level for further processing.
- **NEAR Processing Layer:** The devices of NEAR processing level are typically located at one hop distance from the data sensing devices. This NEAR processing layer also called Edge computing network, comprises





**Figure 1:** The eHealth architecture incorporating the Patient Agent

traditional switches, router, and low profile devices [29]. The devices in the NEAR processing level form a peer-to-peer network. A modified Proof of Stake(PoS) consensus mechanism executes on this peer-to-peer network. Each Fog device runs the similar suits of Blockchain protocols. As a result, they can perform communications between them without the need to trust third parties[2].

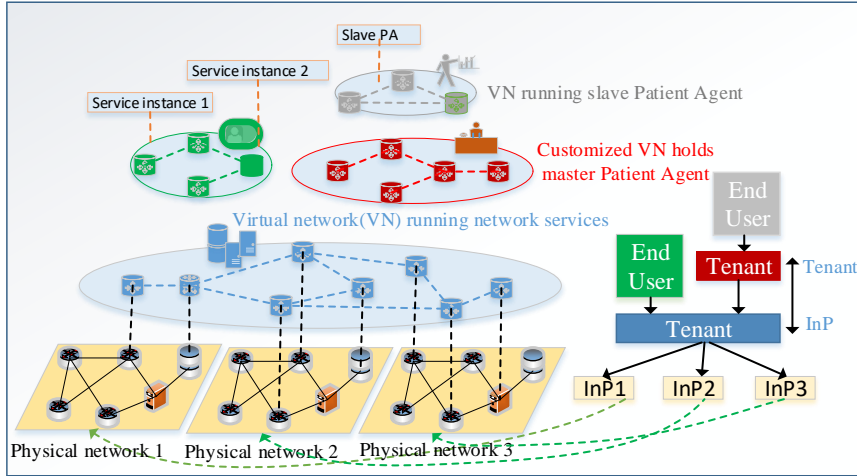
- **FAR Processing Layer:** The FAR processing level includes servers with high computing and storage capabilities. The location of these devices can be far away from the data sensing devices. Cloud servers managed by various proprietary organizations like Amazon, Microsoft, IBM and other stakeholders can provide servers with a large volume of storage and high computing capabilities. Blockchain maintains a distributed tamper-proof ledger replicated amongst multiple nodes. Managing health data characterized as Big data are challenged with the storage in decentralized distributed ledger.

The Cloud servers [61] might support massive storage required for handling decentralized distributed ledger for health data. Multiple instances of a Patient Agent are also deployed in the Cloud to process delay-tolerant and high computing tasks with greater availability and flexibility.

The NEAR and FAR processing layers, each contains  $n \geq 2$  number of instances of a PA and the sensing level has at least  $n \geq 1$  number of PA instances. The Patient Agent hosted on the sensing layer can randomly choose a PA from the NEAR or FAR processing layer every time it has health data to be processed in the upper layers. The smartphone Agent on the sensing level plays the role of master Agent and instructs, one of the NEAR Agents to monitor other NEAR and FAR Agents. The monitoring Agent reports the master Agent if malicious attacks occur on a NEAR or FAR Agent. If an Agent is shut down due to cyber-attacks or suffers from network overload, the master Agent activates a new Agent to take over the role of the infected Agent.

### 3.1. 5G Network Architecture

Logical network slice in 5G technology instantiates on a standard network that comprises terminal access network, core network, access network and transport network. Network resources in 5G are typically defined using two terminologies: infrastructure provider(InP) and tenant. Each InP rents its virtual network resources to the tenant following the business service rules and agreements. The InP can usually host multiple tenants. The tenants can be an infrastructure provider for other tenants. Tenants manage network slices in accordance with the resource demand from different services or applications. Figure 2 depicts the concept of InP and tenant. In Figure 2, three physical networks have been provided by three infrastructure providers(InP). Virtual resources on top of each physical network are allocated to a variable number of tenants. The tenants facilitate network slices to run a wide variety of applications.



**Figure 2:** The 5G network supporting multi-tenancy

Here, we first described a 5G slice architecture designed by ETSI[25]. Next, we explained how the Patient Agent might operate on this 5G network architecture. Figure 3(a) suggests that the resource management in 5G is accomplished in two levels- NFVI level and tenant level. NFVI (Network Function Virtualization Infrastructure) refers to virtualization of hardware resources including computing, storage and networking. Virtual network functions (VNF) that entail distinct network functions such as routing, intrusion detection, domain name service (DNS), caching, and network address translation (NAT) run on these virtual resources.

In NFVI level, VIM directly manages virtualized resources instantiated on the underlying hardware in the form of VMs. VIMs delegate managerial tasks related to network resources to their underlying ICs. The ICs followed by the VIM programmatically manage the network resources and support VM connectivity at the virtualization layer. Like VIM, WIM(Wireless Interface Manager) manages virtual resources related to forwarding instructions or transport layer via WAN IC, which is the SDN controller at NFVI of the transport layer.

SDN (Software Defined Network) controlled Tenant is located on the top of NFVI that provides a tenant with virtual resources. The tenants on top of NFVI manage a set of network slices. Each slice consists of an OSS, a TC (software-defined network controller) and an NSO. The OSS is an SDN application from TC's perspective. The OSS commands TC to create a slice's constituents (VNFs) and logically compose them to realize the network services. The NSO regulates the life cycle of network services and interacts with the TC via OSS. The TC organized as a VNFs depends on the capabilities of the virtual routers, and switches. The TC has two interfaces: northbound and southbound, to interact with end-user and forwarding instructions, respectively. The TC uses southbound interfaces to send composition and forwarding pertinent instructions to virtual routers and switches. The northbound interface of the TC enables users to solicit the required resource capabilities for the network services that they select. Further, the northbound interface enables end users to manage, and operate network slice within limits set by the tenant and retrieve context information such as real-time performance, fault information, and user policies regarding network slices.

The RO, a functional block of a tenant, orchestrates its assigned resources from multiple NFVIs to dynamically satisfying the diverging requirements of network slices. The RO provides each slice with required resources via in-

terfaces of each slice's NSO. The RO(Resource Orchestration) at the tenant level is connected to VIMs of different NFVIs to deliver its assigned resources to the corresponding slices. The tenant can access, reserve, and request virtual resources through RO. VIM and WIM from different NFVIs can interact between them via RO.

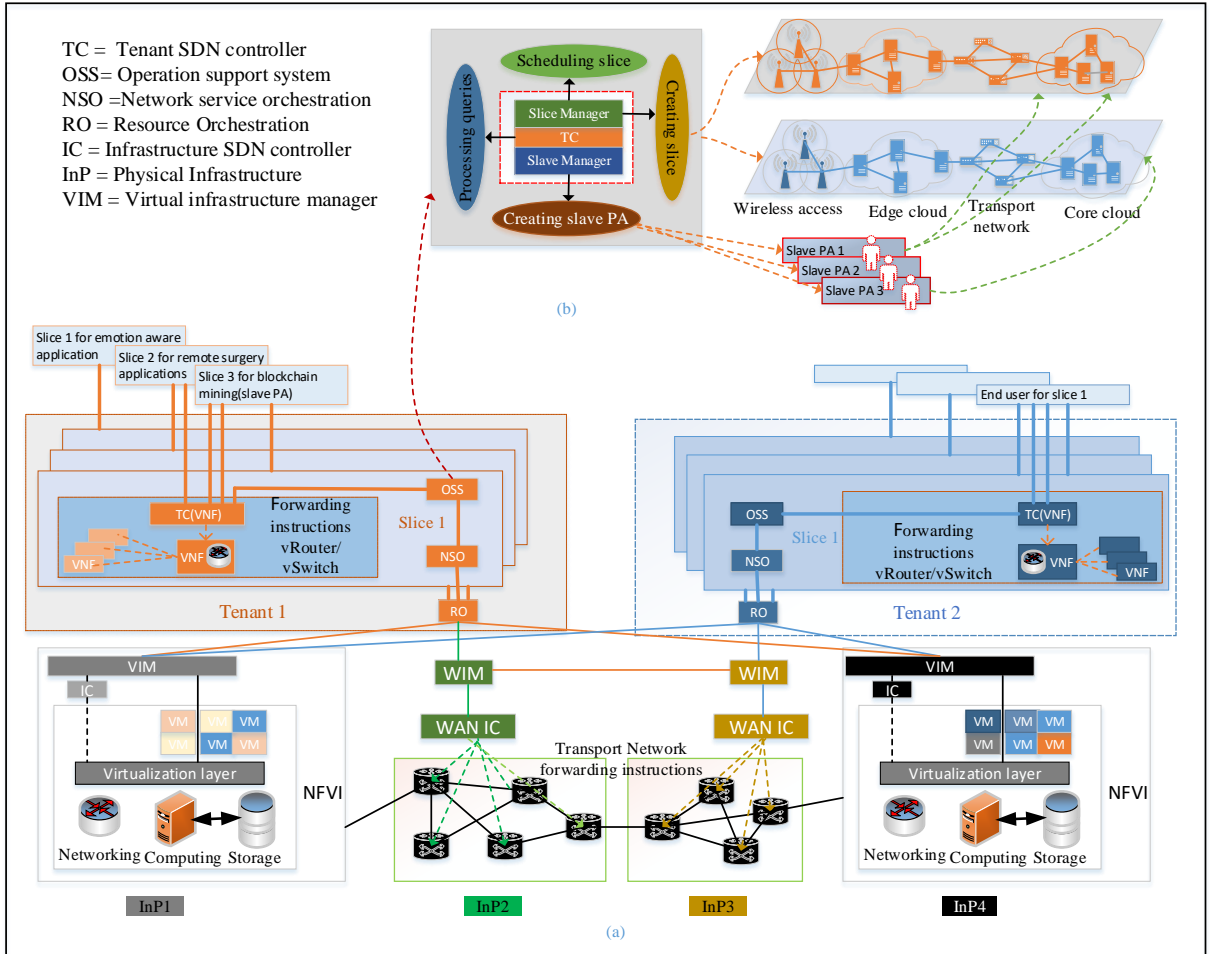


Figure 3: The Patient Agent on ETSI 5G architecture

### 3.2. The Role of the Patient Agent on 5G network

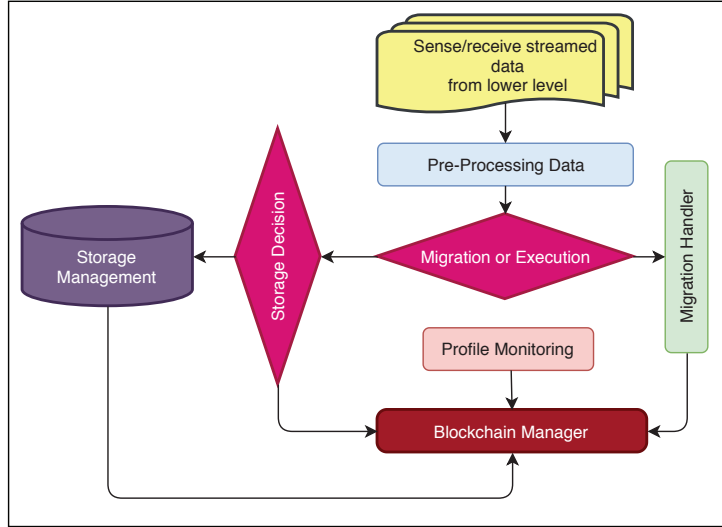
A patient might suffer from multiple complexities and diseases during his or her life span. Handling different medical cases require network slices with various levels of resource capacities to meet patient's QoE and QoS. For instance, resource requirements to serve a patient with diabetes differ from the resource requirements to monitor a patient who suffers from arrhythmia. Therefore, a personalized software Patient Agent is required to continuously record, and analyze health data to assess resource requirements for offering appropriate health services to a patient.

The Patient Agent can fit on the 5G architecture described in section 3.1. We propose to incorporate a master Patient Agent with OSS(Operation Support System) to independently control and manage a diverse range of health applications. The master Patient Agent depicted in figure 3(b), playing the role of a broker or controller, determines the resource requirements for supporting a particular healthcare service. Figure 3(b) depicts that the master Patient Agent comprises two functional components, namely, a slice manager and a slave manager. The slave manager creates a slave Patient Agent dedicated to serve a specific health service. The slice manager asks TC via the OSS to allocate a network slice to run the newly created slave Patient Agent. For instance, the slave manager can assign a slave Patient Agent to run the Blockchain algorithms, including consensus mechanism to a network slice supporting higher computing

because Blockchain technologies require incredibly high computing resources for processing health data in near real-time. Similarly, the slave manager assigns another slave Patient Agent for migrating a task to a network slice with ultra-low speed. A slave Patient Agent for managing remote heart surgery runs on a network slice that has ultra-high bandwidth, very low latency and reliable communication channels. Multiple instances of a slave Patient Agent run on mobile access, Edge and core Cloud of a network slice to make the system fault-tolerant.

### 3.3. Functionalities of the Patient Agent

The Patient Agent supports many health operations including task migration, storage and security management, access control, task execution and Blockchain management. In this framework, few of these significant operations depicted in figure 4 are described below.



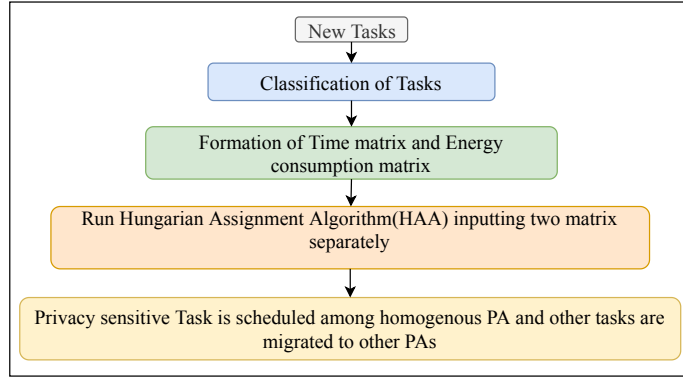
**Figure 4:** The functionalities of the Patient Agent at three levels

#### 3.3.1. Migration Handler(MH)

IoT devices are incapable of providing considerable communication and computation resources required to process a task with low latency. The capabilities of medical IoT devices in terms of processing and storage are less than that of an average computer. Medical IoT devices can save a significant amount of power and make processing faster by offloading heavyweight tasks to their embedded Edge devices[43]. Task offloading in Mobile Cloud Computing(MCC) has been extensively investigated in state-of-the-art research. But, there is still a room to design a secure Blockchain leveraged task offloading mechanism in MEC(Mobile Edge Computing) for the healthcare domain. Maintaining patient's privacy and security during migrating tasks to other remote Patient Agents is indispensable to meet services QoS.

MAUI[62] and CloneCloud[63] determined optimal location(local or remote machine) to execute a task using an offline linear optimization method. They assumed that the energy or time required to offload a task to a machine is known to the system. In this case, the local device needs to collect QoS related information from the remote machine. A remote machine might lie to a local machine, and there needs a secure mechanism to obtain processing capabilities related information of the remote machine. Further, the decision of offloading a task by solving a linear optimization problem involves high computational cost. Further, they statistically partitioned the task offline, which is thought to be non-optimal.

ThinkAir[64] and CADA[65] brought a modification of MAUI's algorithm by considering average execution time based cost required in local device and remote device. They both devised their offloading algorithm for static execution environment, and they assumed these parameters remain unchanged regardless of location or time. Consequently, the proposed methods did not produce an accurate result in a dynamic execution environment. Khoda[66] applied non-linear optimization Lagrange multiplier to decide code offloading. They could improve energy consumption because non-linear Lagrange multiplier's operation is lighter than that of the linear optimization method. Linear regression was



**Figure 5:** The privacy aware task migration process

used to predict the execution time of the offloading task. In healthcare, execution time depends on the data size of the task that varies over time. So, the linear regression-based prediction did not estimate the accurate execution time.

We propose to use a Blockchain to store execution environment parameters of a remote machine. The Blockchain's nodes authorize environment parameters of the potential remote devices that want to take part in executing migrated tasks. The local machine retrieves these parameters from the Blockchain to outsource tasks to neighbouring or remote devices. Further, offloading tasks are categorized based on privacy and time sensitivity to preserve patient privacy and meet QoS, respectively. The Hungarian assignment method that is solvable in polynomial time discovers a set of optimal remote devices for a set of tasks.

**Task Migration Algorithm:** Tasks performed on health data require various levels of resource requirements and security, depending on their complexities, data sensitivity and size. For instance, health data filtration, fusion, compression, and other data mining analyses often require high computing power and massive storage available only on Edge devices or Cloud servers. An early warning module should be executed in Edge devices while streaming data from IoT devices without causing much communication delays that are required to transfer the task to the Cloud server. We consider the facts mentioned above to devise an offloading algorithm.

Figure 5 presents the offloading mechanism of **MH**. A Fog Agent can outsource its tasks to a neighbouring or remote Fog Agent that have spare capacity. In general, a local machine allots a task to a remote machine if any of the following condition is true:

- Condition 1: If energy consumption to execute the task in the local machine is higher than the energy consumption to transmit the task to the remote machine.
- Condition 2: If processing time of the task in the local machine is higher than the processing time of the task in the remote machine.
- Condition 3: Condition 1 and 2

Our task migration algorithm depicted in figure 5 is described as follows. First, tasks are classified before deciding to outsource them. Secondly, two separate matrices are created for each group of tasks. One matrix contains the amount of energy consumption required for a local machine to transmit tasks to the available remote machines. These tasks can be tolerant to some degree of delay without compromising QoS. Another matrix contains response time required to complete the processing of tasks in available remote machines. These tasks are delay-sensitive and need ultra-low latency. Thirdly, privacy-sensitive tasks are scheduled among homogeneous Patient Agents(the agent instances of the same patient) and other kinds of tasks might be scheduled among any Patient Agents regardless of homogeneous or heterogeneous. In both cases, the Hungarian assignment algorithm runs if the number of tasks is  $n \geq 2$ .

It is assumed that a Patient Agent on the smartphone or Fog has multiple tasks( $j = 1, 2, \dots, m$ ) to be assigned to multiple neighbouring or remote agents( $i = 1, 2, \dots, n$ ).

- **Task Classification:** Each task on health data is classified as Privacy Sensitive Task(PST) or Normal Task(NT) following the method described in section 3.3.1. Security-related tasks including encryption/decryption, key



generation, tasks pertaining to sensitive data including HIV/AIDS, sexual preferences, domestic violence are normally deemed as privacy-sensitive tasks. Tasks on low sensitive or not sensitive data, including ECG, body temperature are regarded as normal tasks. Both PST and NT are further classified as Delay Sensitive Task(DST)(heart surgery, emotion aware) or Delay Tolerant Task(DTT) respectively.

• **Time Based Cost Matrix Formation:**

This matrix contains execution time required for all the DSTs(Delay Sensitive Task) to be executed in  $m$  number of available remote Agents. If the execution time of a DST in a remote machine is higher than that of execution time in local machine, a large number is inserted into the corresponding place of that task and agent in the matrix. Consequently, the agent is not chosen for executing that task. The execution time of a task in the local Agent is calculated as follows;

If the local Agent's MIPS(million instructions per second) is  $\mu_l$  and a task( $j$ ) has  $I$  number of instructions including its execution environment, the time needed for the local Agent to complete the task( $j$ ) is estimated as follows:

$$ResponseTime = ExecutionTime + Queue Latency$$

$$T_{j,l} = \frac{I_j}{\mu_l} + \sum_{j=1}^k \tau_{j,l}$$

where the queue latency of the local Agent is  $\sum_{j=1}^k \tau_{j,l}$ . This indicates that the Agent needs to execute  $k$  number of pending tasks besides the current one.  $\tau_{j,l}$  indicates the execution time of a task( $j$ ) that is waiting in the queue of the local Agent.

The aim is to schedule tasks to different remote Agents. The response time from a remote Agent does not depend not only the processing capabilities of the Agent but also the quality of communication link between the local and remote Agents. The response time for a task from a remote Agent is the summation of the propagation time, transmission time, queue delay and execution time required for the remote Agent. The summation of propagation and transmission time is the uploading time. The propagation time is the time for one bit to travel from one router/switch to the next router/switch, and it depends on the distance between two entities and the speed of the communication medium. Transmission time represents the time to get out all the bits of a task from a device to the transmission wire. The response time of a task( $j$ ) from a remote Agent is estimated as follows:

$$ResponseTime = TransmissionTime + PropagationTime + ExecutionTime + Queue Latency$$

$$T_{j,f} = \frac{\omega_j}{\beta_{l,f}} + \frac{d_{l,f}}{v} + \frac{I_j}{\mu_f} + \sum_{j=1}^k \tau_{j,f}$$

where the amount of data involved in the task( $j$ ) is  $\omega_j$ ,  $d_{l,f}$  is the geographical distance between the local and remote Agent,  $v$  indicates propagation speed of the link between the two Agents and  $\mu_f$  represents CPU speed of the remote Agent in MIPS.  $\beta_{l,f}$  is the bandwidth of the communication link between the local and remote Agent. The following matrix for  $m$  number of tasks and  $n$  number of available remote Agents is formed to input it in the Hungarian Assignment Algorithm that discovers optimal allocation of  $m$  number of tasks to  $n$  number of remote Agents in terms of execution time.

$$\begin{pmatrix} T_{1,1} & T_{1,2} & \dots \\ \dots & \dots & \dots \\ \dots & \dots & T_{m,n} \end{pmatrix}$$

$T_{1,1}, T_{1,2}, \dots, T_{1,n}$  represent execution time of task 1 in remote Agent 1 and 2 and so on. Similarly,  $T_{2,1}, T_{2,2}, \dots, T_{2,m}$  represent execution time of task 2 in remote Agent 1 and 2 and so on.

Here,

$$T_{ij} = \begin{cases} T_{j,f} & \text{if } T_{j,l} > T_{j,f} \\ \infty & \text{if } T_{j,l} < T_{j,f} \end{cases}$$

- **Energy Based Cost Matrix Formation:** In case of Delay Tolerant Tasks (DTTs), the local Agent can attempt to save energy consumption by assigning tasks to other neighbouring or remote Agents.

The energy that the local Agent consumes to execute a task ( $j$ ) is estimated as follows:

$E_l = \rho_x \times \frac{I_j}{\mu_l}$ . Where  $\rho_x$  is the power consumption rate of the local Agent while executing a task.  $I_j$  is a number of instructions in the task.

The local Agent consumes energy to transfer a task to a neighbouring or remote Agent. The energy consumption while offloading a task is occurred due to network interfaces and spending idle time of the local Agent (if the agent does not have any tasks in the queue). The energy that the local Agent consumes to offload a task can be estimated as follows:

$$E_f = \rho_d \times T_{j,f} + e_{trans}$$

Where  $\rho_d$  is a power consumption rate of the local Agent during idle mode.  $T_{j,f}$  is the response time of the task( $j$ ) if it is assigned to a remote Agent. The energy consumption due to network interface while transmitting the task can be estimated as follows:

$$e_{trans} = \rho_l \times \frac{\omega_j}{\beta_{l,f}}$$

where  $\rho_l$  is the power consumption rate of the local Agent. The following matrix contains energy consumption of the local Agent for  $m$  number of tasks if those tasks are assigned to  $n$  number of remote Agents.

$$\begin{pmatrix} E_{1,1} & E_{1,2} & \dots \\ \dots & \dots & \dots \\ \dots & \dots & E_{m,n} \end{pmatrix}$$

$E_{1,1}, E_{1,2}, \dots, E_{1,n}$  represent energy consumption of local Agent to execute task 1 in remote Agent 1 and 2 and so on. Similarly,  $E_{2,1}, E_{2,2}, \dots, E_{2,n}$  represents energy consumption of local Agent to execute task 2 in remote Agent 1 and 2 and so on.

Here,

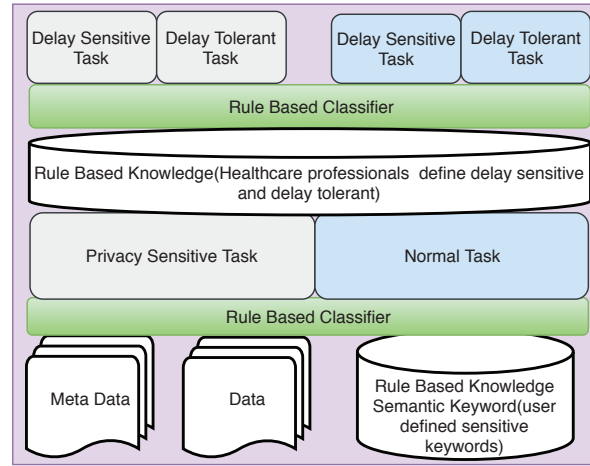
$$E_{ij} = \begin{cases} E_{j,f} & \text{if } E_{j,l} > E_{j,f} \\ \infty & \text{if } E_{j,l} < E_{j,f} \end{cases}$$

**Representation of Hungarian Assignment:** The Hungarian Algorithm is separately run inputting two matrices for two genres of tasks. The mathematical presentation of the Hungarian Assignment on the basis of a time-based cost matrix is shown in (1).

$$\begin{aligned} \min_{t, x} \quad & \sum_{i=1}^n \sum_{j=1}^m t_{ij} x_{ij} \\ \text{s.t.} \quad & \sum_{i=1}^m x_{ij} = 1, \quad j = 1, \dots, m, \\ & \sum_{j=1}^n x_{ij} = 1, \quad i = 1, \dots, n \end{aligned} \tag{1}$$

where

$$x_{ij} = \begin{cases} 1 & \text{if the device is assigned } j^{\text{th}} \text{ task} \\ 0 & \text{if the } i^{\text{th}} \text{ device is not assigned } j^{\text{th}} \text{ task} \end{cases}$$



**Figure 6:** The framework for detecting the sensitivity of a task

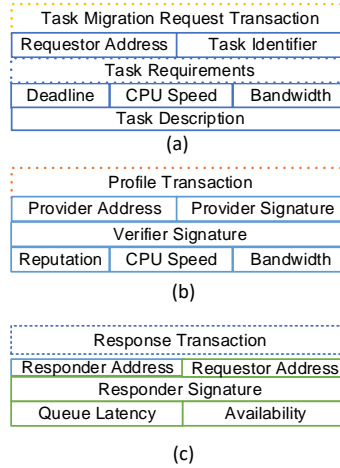
**Identification of Sensitive Tasks:** State-of-the-art research[67, 68, 69, 70] has already addressed the issue of identifying sensitive medical records. Yang[67] presented a model to identify protected health information from the clinical records. The identification method involves the machine learning approach with keyword-based and rule-based techniques to separate protective health terms. Jindal et al. [68] presented semi-supervised techniques to detect sensitive words from clinical narratives. They used the SNOMED CT to train the model, which eliminates the needs of an annotated and unannotated dataset. Authors also applied a rule-based method to classify the negotiation words and experienter. Sanchez[69] proposed an automatic sensitive terms detection system using an information-theoretic approach. They used the web-based corpus to make the solution more generalized and domain-independent. Tesfay[70] presented an architecture for assessing user-centred privacy risk. The architecture includes three components: privacy detection system(PDS), risk communication manager(RCM), and privacy quantification(PQ). RCM determines the privacy level of personal data like high, medium and low sensitive. PQ quantifies the privacy risks based on probabilistic and combinational techniques.

Here, we used a sensitive content identification framework designed following[67, 68, 69]. The framework automatically categorizes sensitive health data for MH(Migration Handler) because tasks on the sensitive data is considered to be sensitive. The framework depicted in Figure 6 has two kinds of the rule-based classifier. The data and metadata containing task information are fed into the first rule-based classifier. The classifier applies the rules stored in the Rule Base Knowledge(RBK). The RBK stores the user's feedback and expert's feedback regarding the sensitive keywords in health data. The first rule-based classifier categorizes the task as Privacy Sensitive Task(PST) or Normal Task(NT) using RBK. Next, the task(PST or NT) is classified by the second rule-based classifier as Delay Sensitive(DST) or Delay Tolerant Tasks(DTT). The second classifier applies the rules from another rule-based knowledge formed with the feedback collected from the healthcare professionals.

### 3.3.2. Profile Monitoring(PM)

The Migration Handler of a local Agent aims to outsource and distribute computing tasks to neighbouring and remote Patient Agents that have additional computing resources. Each Patient Agent, therefore, needs to know the profile of other Agents on the peer-to-peer Edge network. The Profile Monitoring(PM) module requires queue latency, CPU speed, availability, and bandwidth information of the available remote Agents to pass on to the Migration Handler. The malicious Agents might lie to a client Agent about their performance parameters. To address this issue, we propose to utilize Blockchain to manage and store performance parameters for migrating tasks. Every Patient Agent registers their performance parameters on the Blockchain.

The PM creates several transactions (depicted in figure 7) related to task migration. For instance, a Patient Agent issues a Profile Transaction(PT) containing CPU processing, storage capacity and other parameters including network capacity when it first joins in the Blockchain. Miner nodes on the Blockchain verify these parameters packed in the transaction. A Patient Agent might include additional resource capabilities over time. If it does so, it needs to make a



**Figure 7:** The transactions for migrating tasks

new Profile Transaction in the Blockchain. In this case, Blockchain nodes also process, and validate these transactions. The PM of a PA broadcasts a Task Migration Request Transaction (shown in figure 7(a)) throughout the peer-to-peer Blockchain network when it needs to outsource tasks. The other Agents with available resources reply to the PM by making a Response Transaction (shown in figure 7(c)) that contains the dynamic resource information such as queue latency of the remote Agent. The local Agent can retrieve static profile information of that Agent from the Blockchain.

### 3.3.3. Execution Unit(EU)

The execution unit(EU) of the PA performs the processing of health data. The processing might include filtration, fusion, generating a warning, automatic diagnosis and other operations. A Patient Agent has the option to choose an EU among its own EU, other Patient Agent's EU and smart contract-based EU. A smart contract refers to a set of rules encoded using a specific programming language[71]. Every Blockchain node stores coded rules for a smart contract. A Smart Contract is triggered when a transaction specified to that smart contract is issued in the Blockchain network. The Patient Agent might make different kinds of smart contracts for processing a patient's health data. The followings are a few examples of the smart contract.

- **Smart Contract for Registration:** This contract is executed once a Patient Agent registers in the Blockchain for the first time.
- **Smart Contract for Data Filtration:** The contract contains code for clinically uninteresting health data filtered out.
- **Smart Contract for Data Classification:** The contract holds the procedure for categorizing health data as normal and abnormal. The contract is triggered upon the request of data classification.
- **Smart Contract for Warning Generation:** The contract holds the code to generate alarm after analyzing continuously streamed medical data.
- **Smart Contract for Task Migration:** This contract is triggered while migrating tasks to high computing devices.

### 3.3.4. Storage Determination(SD)

Health-related data can be stored on diverse repositories including government-managed repositories (e.g. myGov electronic health record in Australia), Blockchain, on healthcare service provider servers, on private Cloud servers, on a patient's personal computer or any other devices. Different repositories provide a different level of security and privacy. Patients have diverse privacy preferences. The SD will model a patient's privacy preferences and experts knowledge of security to automate decisions regarding preferred storage for health data. This module will determine the storage repositories for data stream rapidly from wearable sensors and other kinds of health data.

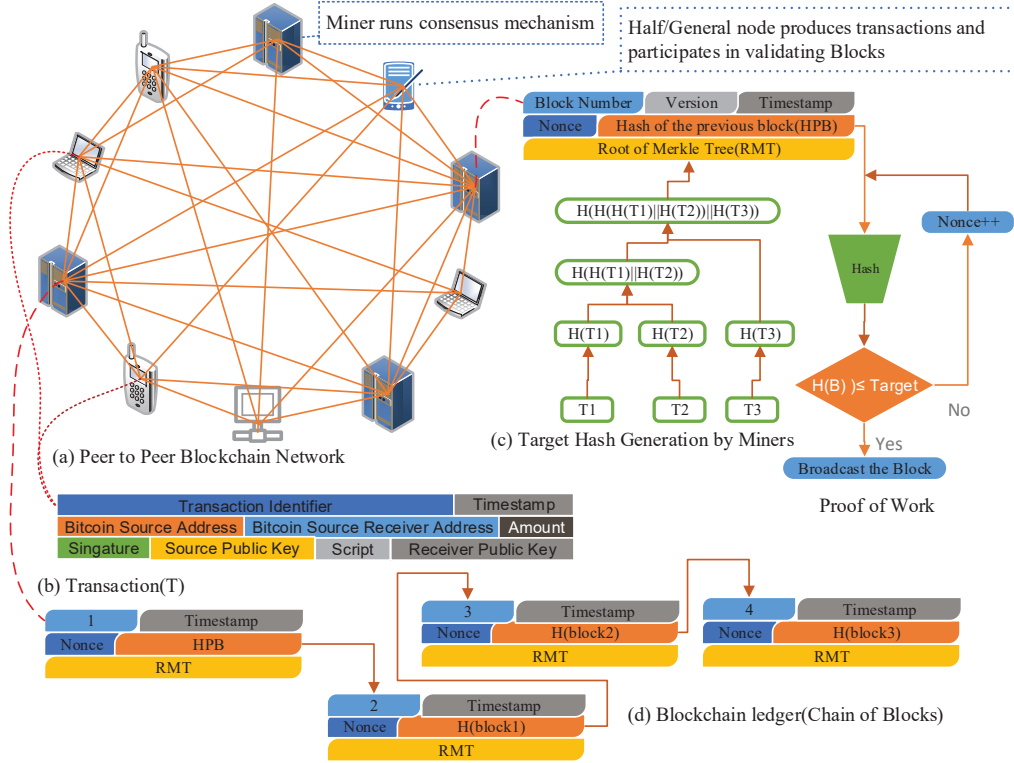


Figure 8: The Bitcoin Blockchain

### 3.3.5. Blockchain Manager(BM)

Blockchain's structure offers a couple of advantages such as tamper-proof storage, patient's privacy and processing data without the need to trust third parties. The Blockchain maintains a unique ledger replicated amongst multiple nodes. The ledger is formed with a series of confirmed Blocks connected between them in linked list fashion where Block comprises a certain number of transactions packed in a secure Merkle tree. Blockchain nodes add a new Block in the current ledger by running a consensus mechanism.

The BM can apply Blockchain technologies to perform task migration, store streamed health data, operate diagnosis, and control access. The BM also undertakes security services such as key management, encryption/decryption of health data, make various kinds of health data transactions to be inserted into the Blockchain, and participates in consensus protocol that is required to add a new data Block in the Blockchain. In this section, we describe a lightweight modified Proof of Stake consensus mechanism for our IoT healthcare architecture.

The components of a standard Blockchain(used in Bitcoin) are illustrated in figure 8 before discussing our customized Blockchain.

1. The Blockchain operates on a peer-to-peer network depicted in figure 8 (a). Nodes of this network is categorized into three groups: half nodes, general nodes and Miner nodes.
2. The half node/public node produces transactions formatted as in figure 8(b) and broadcast throughout the peer-to-peer network.
3. The Miner nodes collect a certain number of transactions and pack them into a Merkle tree to create a Block. Each Miner repeatedly inputs the Block into a cryptographic hash function incrementing the nonce field by one every time as long as the hash function produces a target hash code for the Block. This process depicted in figure 8(c) is called Proof of Work(PoW). Only one Miner which first publishes the Block with Proof of Work receives the reward for doing this.
4. Finally, all nodes except the half-node link the Block to the end of the existing chain as depicted in Figure 8(d) by running a verification process.



**The Lightweight Consensus Mechanism:** In our eHealth architecture, the smartphone transmits its captured health data to NEAR processing devices(Fog devices). The NEAR processing level should execute the Blockchain operations to process medical IoT data in near real-time. But, Blockchain technologies requires high computational cost and storage, and NEAR processing devices do not have appropriate capabilities to accommodate a Blockchain. However, Blockchain technologies can be adapted in an IoT healthcare if Blockchain operations are partitioned. We can allot Blockchain operations to three layers of the IoT healthcare architecture, considering devices' capabilities of these layers. For example, the smartphone Agent in the sensing layer can define the structure of transactions and initiates data flow. The Fog Agents in the NEAR processing layer can execute a lightweight consensus mechanism and store metadata about a Block. This reduces delay in Block's confirmation on the Blockchain because NEAR processing devices are located at one hop from the medical sensors. The Agents in the FAR processing layer can permanently provide storage for the Blockchain-based health ledger. The Cloud Agents insert Blocks into Cloud ledger after the Fog Agent confirms the Block by running a consensus mechanism.

The Proof of Work(PoW) used in Bitcoin and Ethereum is the most decentralized consensus method. PoW is employed to solve the byzantine general's problem. In PoW, the Miners write a new Block in the existing chain of Blocks through generating a target hash of the new Block. The Miners compete to come up with the target hash, and the winner obtains a certain amount of reward. Every node ensures containing an identical version of the newly constructed Block in the Blockchain through a validation process. The PoW protects the Blockchain from DoS, and Sybil attacks as attackers are not being encouraged to invest a tremendous amount of computational resources required for the PoW. But, this consensus mechanism is not suitable for processing streamed medical IoT data because the medical data transactions necessitate faster processing to meet patient's QoS. The Bitcoin network generally requires approximately 10 minutes [14] to reach on the agreement for a Block due to using PoW mechanism.

One of the most proficient consensus methods is Proof of Stake (PoS)[14] in terms of scalability and processing time. In this process, prospective Miners have to lock their coin to the system. A Miner having higher share makes the next Block, and it receives incentives for this. Delegated Proof of Stake(DPoS) is a variation of standard PoS. With DPoS, users or a group of delegates select a set of witness nodes through a voting mechanism[72]. The weight of a delegate's vote for a witness node is proportional to the amount of the witness node's deposited coin in the system.

A witness obtaining the maximum number of vote from the panel accumulates the transactions and organizes those into a Block. Other witness nodes verify the newly created Block to confirm it in the Blockchain. Some drawbacks of this mechanism are: few nodes can dominate the entire network, which makes the system vulnerable to a 51% attack. Nodes having a high stake can influence the Blockchain network more than that of nodes with a small stake due to assigning the weight of a vote based on the stake a node holds. Further, delegates might collude to vote for a particular group of witness nodes. We propose a modified PoS to mitigate some of the standard PoS drawbacks mentioned above. The section below describes the modified consensus mechanism demonstrated in Algorithm 1.

- **Cluster Formation:** In the NEAR processing layer, Fog/Edge Agents within a certain geographical range( $R$ ) form a cluster. The Figure 1 depicts three clusters:  $C_1$ ,  $C_2$ , and  $C_3$  at the NEAR processing level. Every cluster has a random number of Patient Agents, where one member is elected as cluster head. The cluster head( also called leader) participates in running the consensus protocol of the Blockchain by locking a certain amount of stake to the system. Depositing a certain amount of digital coin is mandatory for every Miner. The node identified as malicious one loses its stake. The node also receives a negative review from peer Miners so that it has a slim chance to be a Miner next time.
- **Leader (Cluster Head) Nomination:** A cluster head(CH) is selected from each cluster considering multi-criteria of the member nodes. The selection criteria include a node's performance parameters, reputation, and the amount of stake. These criteria are combined using a Fuzzy Inference System(FIS) to estimate a fitness value. Every node's information regarding the mentioned criteria is recorded in the Blockchain and can be retrieved from the Blockchain when they are needed.

The performance parameters include the processing speed of an Agent's device, memory capacity, availability, distance coefficient of variation, and transmission delay. Here, processing capabilities in MIPS, memory capacity and availability are symbolized as  $p_1$ ,  $p_2$ , and  $P_3$  respectively. These parameters are normalized in the range from [0 to 1] and then those normalized values are summed up as follows:

$$\gamma = \sum_{p=1}^3 \frac{1}{(1 + e^{-p_i})}$$

A node whose distance from the other nodes in a cluster is uniform should be selected as a cluster head. The reason is that a cluster head which is closer to most of the member nodes and a bit far away from few other member nodes might experience an inconsistent delay to receive/send data from/to all the member nodes. A node with almost uniform distance from other nodes is appropriate as a cluster head for synchronizing timestamp between nodes. Distance coefficient of variation( $CoV$ ) can be applied to estimate the consistency of a node with respect to distance from other nodes in a cluster. If  $d_1, d_2, \dots, d_n$  are distance of other nodes from a node  $i$ ,  $CoV_i$  is calculated as follows.  $CoV$  is the result of standard deviation divided by the mean of a set of values.

$$\text{If the mean } \mu_i = \frac{\sum_{k=1}^n d_j^k}{n} \text{ and standard deviation } \sigma_i = \sqrt{\frac{1}{n} \sum_{k=1}^n (d_j^k - \mu)^2} \text{ then Coefficient of Variation } CoV_i = \frac{\sigma_i}{\mu_i}$$

Next, delay refers to the time required for one bit to send from the source to destination. Here, the harmonic average delay concerning a node( $i$ ) refers to a harmonic average of the amount of delay to receive one bit from other nodes in the cluster.

$$H_i(t) = \frac{n}{\sum_{j=1}^n \frac{1}{t_j}}, \text{ where, } j = 1 \text{ to } n \text{ member nodes need } t_1, t_2, \dots, t_n \text{ respectively to send one bit data to a particular node } i.$$

The harmonic average delay is normalized in the range [0 to 1] as follows.

$$\tau_i = \frac{1}{1 + e^{-\frac{1}{H_i(t)}}}$$

The higher CPU speed, memory capacity, availability of a node and the lower average delay and coefficient of variation with a Miner, the better the Miner is. Therefore, each node in the cluster combines its performance as follows.

$$P_i = \frac{1}{(1 + e^{-\frac{\gamma}{\tau \times CoV_i}})}$$

The second criteria for selecting a cluster is reputation. An Agent receives a transaction containing a positive or negative reputation in the range of [1 to 5] when the Agent serves a requestor's service. A requestor can pick a value from this range [1 to 5] as a positive or negative reputation on the basis of QoS including timely service, accuracy and others an Agent offers. If an Agent serves multiple services from multiple requestors, it obtains a reputation transaction for every service it offers. Blockchain records each reputation an Agent receives. The positive or negative reputation that a requestor provides to an Agent is multiplied by the requestor's positive reputation and divided by the negative reputation of the service providing Agent. The initial positive or negative reputation of each Agent is assumed 1.

It is supposed that an Agent( $i$ ) already obtained positive reputation from  $n$  number of service requestors  $\alpha_1, \alpha_2, \dots, \alpha_n$  and negative reputation  $\beta_1, \beta_2, \dots, \beta_n$ . The positive reputation of these service requestors is  $\omega_1, \omega_2, \dots, \omega_n$ . The ultimate reputation for a service providing Agent while nominating cluster head is estimated as follows.

$$r_i = \sum_{j=1}^n \frac{\alpha_j \times \omega_j}{\beta_j}$$

The aggregated normalized reputation is as follows:  $R_i = \frac{1}{1 + e^{-r}}$ .

The third criteria named digital coin( $c$ ) that an Agent( $i$ ) has in the system is normalized as follows:  $S_i = \frac{1}{(1 + e^{-c})}$ .

Now, every Agent in a cluster calculates their fitness( $f_i$ ) using the criteria outlined above. A Fuzzy Inference System(FIS) is used for this because fuzzy rules can represent sophisticated heuristics more appropriately than crisp rules. The input of the FIS is an Agent's performance, reputation and the current stake. An Agent with higher fitness is delayed for a short period before declaring itself cluster head. The member node in a cluster

waits for the following period presented in (2)

$$T_i = \Delta T \times \left(1 - \frac{f_i}{\sum_i f_i}\right) \pm \lambda \quad (2)$$

Where  $\Delta T$  is the time interval for electing cluster head, and  $\lambda$  represents a short random time duration that is used to differentiate waiting time for the Agent having the same fitness. The Agent that expires its waiting time broadcasts its identifier throughout the cluster. The other cluster members verify the estimated fitness of the Agent and acknowledge their approvals for this Agent. The Agent finally wins as a leader of the cluster only if it obtains specific numbers of approvals. Once a leader makes a Block, a specific value is deducted from its original fitness so that the chance of other node's being leader increases next time.

- **Super Leader Nomination:** A super Leader is randomly selected from the selected set of cluster heads. This super leader is responsible for making new Block packing a certain number of transactions from the transaction pool. A new super leader is elected after the current super leader prepares a certain number of Blocks. A new round begins when every cluster head eventually becomes super leader. Every cluster head takes part in verifying a new Block before broadcasting it throughout the network. The Fog Agent stores the metadata about a Block and transmits the complete Block to FAR processing layer for the permanent storage. Half of the reward for mining is awarded to the super leader, and the rest of the half reward is equally divided among the cluster heads. The super leader selection is as follows: each cluster head except the Agent that was already elected as super leader generates a random number between 0 and 1 according to (3). A cluster head becomes a super leader if its random number is less than the threshold stated in (3) and obtains the threshold number of votes from the other cluster heads.

$$I = \begin{cases} \frac{p}{1-p[r \bmod (1/p)]}, & N \in G \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

where  $p$  is the percentage of cluster head in the Fog network,  $N$  is a total number of cluster head, the  $r$  is the number of rounds of selection, and  $G$  is the set of cluster heads that have been elected as super leader in round  $\frac{1}{p}$ .

Every node in a cluster can participate in the proposed PoS by turns. This consensus mechanism is less vulnerable to 51% attack than DPoS as a leader comes from each cluster. Unlike PoS, the rich node has less chance to be a leader of a cluster because the cluster head is not only selected based on the locked coin but also reputation and performance parameters. Further, the reduction of specific points from the current leader's fitness prevents the node from being a leader for the subsequent round.

*Fuzzy Inference System(FIS) to assess a node's fitness* A fuzzy expert system is a collection of fuzzy rules and membership functions that are used to reason about data. Fuzzy inference process that refers to a process of mapping a given input to  $n$  output by using the theory of Fuzzy sets. The Fuzzy inference process involves four steps: Fuzzification, rule evaluation, aggregation, and defuzzification. The functional blocks of the FIS depicted in Figure 9 to generate fitness used in the consensus algorithm is briefly described below:

- **Fuzzifier:** The first step of fuzzy inference is to map crisp(numerical) inputs into degrees to which these inputs belong to respective fuzzy sets. Fuzzifier converts crisp inputs to linguistic variables applying membership functions such as triangular, trapezoid or Gaussian functions. Figure 10 shows the conversion of crisp input(performance) using MATLAB FIS. The numerical value in the range of [0 to 1] is expressed in a linguistic variable: low, medium and high.

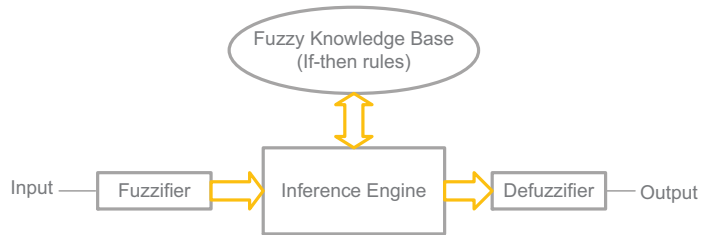
**Algorithm 1:** Modified PoS Consensus Protocol

---

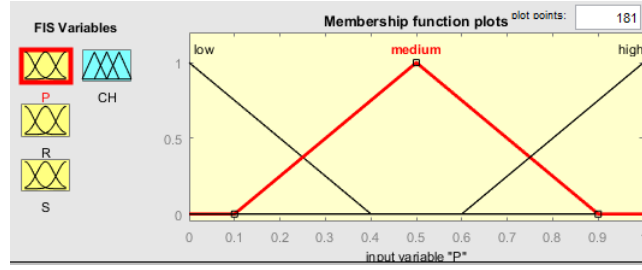
**Data:** Performance Transaction(PT), Reputation Transaction(RP), Stake Transaction(ST), Agent number(  $n_k$ ) in a cluster  
**Result:** a set of healthy Miners

- 1 Every Fog Agent generates  $PT_i$ , and  $ST_i$  for themselves and obtains  $RP_j$  from service providers ( $j$ ).
- 2 Form clusters with Fog nodes within a threshold range( $R$ ).
- 3 **for** each cluster  $k = 1$  to  $l$  **do**
- 4     **while** leaderElection = true **do**
- 5         **for** each member Agent  $i = 1$  to  $n_k$  of a cluster **do**
- 6             Extract parameters from  $PT_i$ ,  $LST_i$ ,  $RP_i$  to produce  $P_i$ ,  $R_i$ , and  $S_i$
- 7             
$$P_i = \frac{1}{(1 + e^{-\frac{\gamma}{r \times CoV_i}})}$$
- 8             
$$r_i = \sum_{j=1}^n \frac{\alpha_j \times \omega_j}{\beta_j}$$
- 9             
$$R_i = \frac{1}{1 + e^{-r}}$$
- 10            
$$S_i = \frac{1}{(1 + e^{-c})}$$
- 11             $f_i \leftarrow \text{fuzzyInferenceProcess}(P_i, R_i, S_i);$
- 12         **end**
- 13         
$$T_i = \Delta T \times (1 - \frac{f_i}{\sum_i f_i}) \pm \lambda$$
- 14         Every member node in the cluster( $j$ ) sets their timer( $T_i$ ).
- 15         **if**  $T_i$  is expired **then**
- 16             Broadcast  $nodeId$  throughout the cluster for approval.
- 17         **end**
- 18         **if**  $\text{approvalCount}[nodeId] \geq \frac{2}{3} \times n_k$  **then**
- 19              $leader_j \leftarrow nodeId;$
- 20              $f_{nodeId} \leftarrow f_{nodeId} - \epsilon$
- 21              $leaderElection \leftarrow false$
- 22         **end**
- 23     **end**
- 24 **end**
- 25  $superLeader \leftarrow \text{selectSuperLeader}(leader_1, \dots, leader_m)$

---

**Figure 9:** The FIS for determining node's rate

- **Inference Engine(Rule evaluation and aggregation):** Inference engine stores fuzzy rules. These rules consist of "If" and "then" statement. This step takes the fuzzified inputs and applies them to the antecedents of the fuzzy rules. Few "If-then" rules for evaluating fitness are represented in Table 2. In Figure 11, **P**, **R** and **S** stand for miner's performance, reputation score and the amount of stake receptively.
- **Defuzzifier:** Finally, Defuzzifier converts the fuzzy outputs generated by Inference Engine to a single crisp number using Centre of Gravity (COG) or other methods such as the centre of Area(AOC), bisector of area(BOA). A fuzzy output is generated from each "ifthen" rule firing. Each output fuzzy is the input of the Defuzzifier. The



**Figure 10:** The membership function for performance parameter

**Table 2**

The ifthen rules for the FIS

SL. No.	Rules
1	if(P is low) and (R is low) and (S is low) then (CH is low)
2	if(P is low) and (R is medium) and (S is low) then (CH is low)
3	if(P is low) and (R is high) and (S is medium) then (CH is medium)
4	if(P is medium) and (R is low) and (S is medium) then (CH is low)
5	if(P is high) and (R is high) and (S is medium) then (CH is high)
6	if(P is medium) and (R is high) and (S is medium) then (CH is high)
7	if(P is high) and (R is medium) and (S is high) then (CH is medium)
8	if(P is high) and (S is high) then (CH is medium)
9	if(P is low) and (S is high) then (CH is low)
10	if(R is high) and (S is high) then (CH is high)

Defuzzier aggregates fuzzy output set into a crisp out as depicted in Figure 11 where multiple rules have been fired. Figure 11 illustrates that if  $P(\text{performance score}) = 0.476$ ,  $R(\text{reputation score}) = 0.608$  and  $S(\text{score from deposited coin}) = 0.331$ , the aggregated score using (4) is 0.805( indicates the probability of the node being leader).

$$COG = \frac{\int_a^b \mu_A(x) x dx}{\int_a^b \mu_A(x) dx} \quad (4)$$

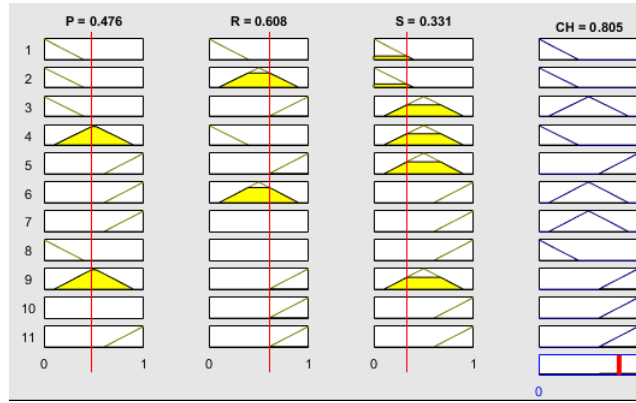
**Data Block Structure:** Health Data Block is presented in Table 3. The Block is divided into two parts- header and contents. The header holds metadata for contents.

**Block Validation:** The leaders selected by Algorithm 1 verify a new Block before sending the Block to the FAR processing layer. The verification process is depicted in Algorithm 2 where leader Agents check the hash value of the immediate previous Block, the integrity and signature of all the transactions packed in Merkle Tree of the Block. The leader first broadcasts the Block throughout the NEAR processing layer. If specific numbers of leader certify the Block as a valid Block, Agents in the NEAR processing layer sends the Block to the Agents in the FAR processing layer for permanent storage. The Agents in the NEAR processing layer store metadata about a confirmed Block.

### 3.4. Security Protocol for the Decentralized Patient Agent

The same Agents replicated at the NEAR processing layer, and FAR processing layer require a secure communication channel to transfer health data between them. In this IoT eHealth, we used some standard security protocols to enforce security for the Patient Agents. The security protocol for eHealth architecture is described below.





**Figure 11:** The output of the Defuzzifier

**Table 3**

The Format of Data Block

Block Header of Blockchain	
Field	Description
Version	Block Version Number
Block Types	It indicates diverse types of Blocks(health record, financial record, diagnosis record)
Previous Block Hash	This field contains hash code of the previous confirmed Block
Timestamp	This field records the Block creation time
Merkle Tree Root	Transactions are packed into a Merkle tree and this field stores the root of the tree
Vote	Miner verifies the Block and votes

### 3.4.1. Digital Signature

The Bitcoin or Ethereum Blockchain uses the PKI(Public Key Infrastructure) as an address for a user or Miner. Blockchain users generate a digital signature using PKI. The PKI can preserve a node's pseudonymous properties, but a user's transactions can be correlated to each other while the Miners verify those. In eHealth, a patient's privacy has the risk of being compromised if an attack can connect his sensitive health data to an Agent of the patient.

Ring Signature[73] can be an alternative to traditional PKI digital signature to preserve a patient's privacy. A transaction's owner can form a digital ring signature by merging a group of other users' signature. As a result, an attacker cannot identify the owner of the data transaction because multiple entities participate in forming a ring signature. In our eHealth, the Patient Agent executing in the sensing layer is the signer and other neighbouring Agents at the NEAR processing layer are the ring members.

The Ring Signature's format depicted in Figure 12(c) is represented as  $(m, P_1, P_2, \dots, P_r; v, x_1, x_2, \dots, x_r)$  where  $P_{1 \leq i \leq r}$  is the public key of the ring members, and  $x_{1 \leq i \leq r}$  is a random number selected by the signer(the Patient Agent in smartphone/Fog) for  $P_{1 \leq i \leq r}$ .  $m$  is signified as the original message for which the digital signature needs to be generated and  $v$  is the generated message(digital signature) that needs to be verified. A typical Ring Signature is generated as follows;

- **Message Digest:** The signer(data owner) generates  $k = H(m)$  and a random value( $u$ ). The signer performs a symmetric encryption on  $u$  with key( $k$ ) to produce  $v = Enc(k, u)$ .
- **Signature Merge:**  $e = x_i^{P_i} (mod N_i)$  is calculated for each ring member except the signer. Here,  $x_i$  is a random number picked by the signer for the  $i^{th}$  member and  $P_i$  represents the public key of the member nodes. The signer also produces  $v = v \oplus e$  for each member. The signer calculates  $x_s = (v \oplus u)_d (mod N_s)$  where  $d$  is the secret key of the signer.
- **Complete Signature:** Finally, the signer forms the signature as  $(m, P_1, P_2, \dots, P_r; v, x_1, x_2, \dots, x_r)$

**Algorithm 2: Block Validation Procedure**


---

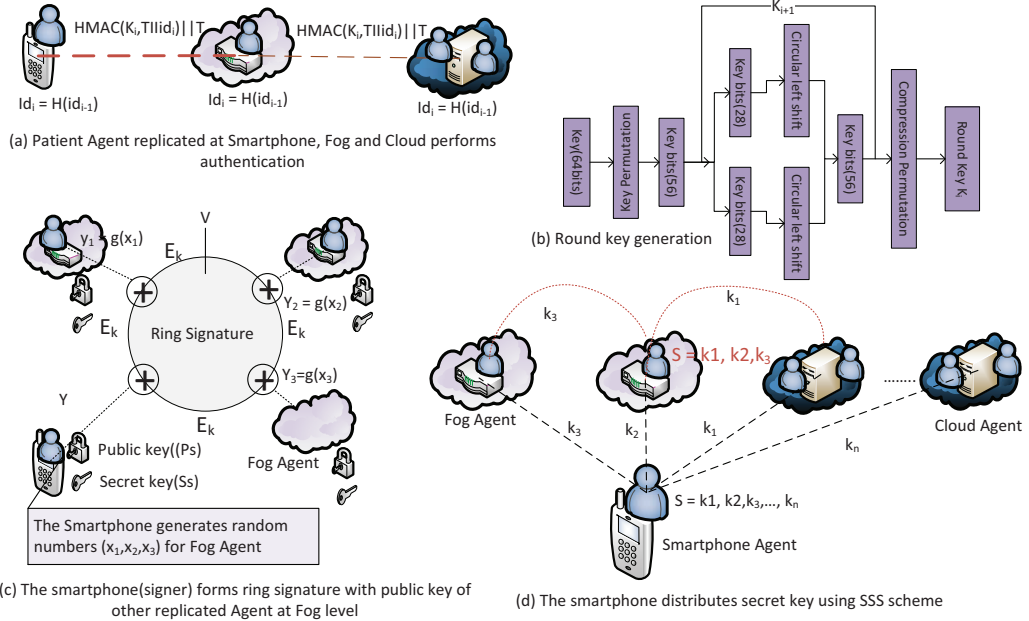
**Data:** Block(B)  
**Result:** Confirmed Block

```

1 initialize  $statusSig = false$ ,  $statusContent = false$ ,  $statusPrevBlockHash = false$ ,  $countVote = 0$ 
2 super leader organizes a certain number of transactions into a Block
3 super leader sends the Block to leaders for the approval
4 for each leader  $k = 0$  to  $l$  do
5   while  $newBlockRequest = true$  do
6      $statusSig \leftarrow verifySignature(B)$ ;
7      $statusContent \leftarrow verifyIntegrity(B)$ ;
8      $statusPrevBlockHash \leftarrow verifyPrevHash(B)$ ;
9   end
10  if  $statusSig == true \wedge statusContent == true \wedge statusPrevBlockHash == true$  then
11     $countVote++$ 
12  end
13 end
14 for each general Edge Agent  $i = 1$  to  $m$  do
15   if  $voteCount \geq thresholdCount$  then
16     Block is transferred to the Cloud Blockchain
17   else
18     Block is rejected
19   end
20 end

```

---

**Figure 12:** The security method for the framework

The health data transactions to be processed in the Blockchain contains a ring signature. The Blockchain nodes check the data integrity using the ring signature. But, Blockchain Miners cannot trace the transaction's creator and distinguish the creator from other ring members.

### 3.4.2. Authentication between replicated Patient Agent

The Patient Agents at the three levels need to perform authentication using a session key every time they exchange medical data. The replicated Patient Agents (homogenous Agents) in the smartphone, Fog and Cloud dynamically come up with the same session key using DES Round Key generation algorithm [74] depicted in Figure 12(b) to avoid a man in the middle attack that occurs during the exchange of session key. A primary key for generating session key is inserted into the replicated Patient Agent during installation. Later, the generation of the session key is achieved by using the same algorithm. Suppose,  $K_s^i$  is the session key generated by the three replicated Patient Agent for the  $i^{th}$  session. The homogeneous Patient Agents need to exchange  $HMAC(K_s^i, Time || Id_i) || Time$  depicted in Figure 12(a) to authenticate each other. Key Exchange between heterogeneous replicated Patient Agents occurs using Diffie-Hellman key exchange method [75].

### 3.5. Data Encryption Key Management

Symmetric key encryption technique such as AES or DES [74] is appropriate for memory and power-constrained devices like smartphones [15]. The Patient Agent replicated at three levels: smartphone, Fog and Cloud each store asymmetric data encryption key or exchange the key. The data encryption key might be compromised by a malicious hacker or by a rogue Fog or Cloud administrator. The replicated Patient Agent needs to secure their keys. There needs a key management protocol that does not allow any Patient Agent to obtain the key without the approval of other replicated Patient Agents. The SSS (Shamir's Secret Sharing) scheme depicted in Figure 12(d) is utilized to distribute a secret key among the replicated Patient Agents. The secret key ( $S$ ) to decrypt data is divided into pieces of data  $S_1, \dots, S_n$  and each piece is shared with  $n$  replicated Patient Agent accordingly.

1. A Patient Agent requires knowledge of  $k$  or more  $S_i$  from other replicated Patient Agent to compute the complete secret key  $S$ . For instance, if the Patient Agent is replicated in five different devices,  $k$  might be two or three.
2. A Patient Agent cannot reconstruct the secret key  $S$  with fewer than  $k$  pieces.  $S$  remains completely undetermined with knowledge of  $k - 1$  or fewer  $S_i$  pieces

This key sharing scheme is called  $(k, n)$  threshold. Every time, a Patient Agent requires to decrypt data, it asks  $k - 1$  pieces of  $S_i$  from other replicated Patient Agent to make the complete secret key ( $S$ ). This scheme prevents the attack from gaining unauthorized access to keys, even if the device is compromised.

## 4. Performance Analysis

In this section, we discussed and analyzed the performances of the key algorithms of the proposed architecture. The simulation is coded using Java Programming following iFogSim [76]. Table 4 presents the simulation parameter.

**The Consensus Mechanism:** The nodes in the simulated are located in  $1000 \times 1000m^2$  area. Performance of the consensus mechanism is estimated considering a variable number of nodes 100, 200, 300, 400 and 500 and a variable number of clusters such as 5, 10, 15, 20, 25, 30, 35, 40, 45 and 50 respectively for each group of nodes. The member nodes within a cluster can directly send or receive data to/from a cluster head. But, nodes within the inter-cluster communicate using the shortest path routing such as Dijkstra's algorithm. The performances of the modified PoS consensus mechanism and the standard PoS are investigated for the following parameters.

- **Energy Consumption:** Energy consumption refers to the energy required for transmitting, receiving transactions, validating a certain number of Blocks on the simulated network.
- **Block Generation Time:** This refers to the time required for transmitting, making Blocks and validating a certain number of Blocks on the simulated network.

The modified PoS consensus mechanism is executed ten times in the simulated network, and the performance graphs are depicted with average values generated from 10 execution runs. The standard PoS runs on a horizontal network, and the modified PoS is designed to run on a hierarchical network. For both kinds of consensus mechanisms, nodes that lock digital coin to the system participate in mining. Figure 13 depicts the consumption of energy and execution time to generate 100 number of Blocks providing that a variable number of nodes and clusters are considered.

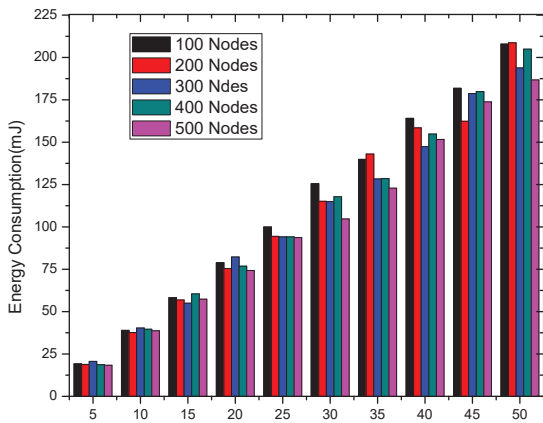
The graph depicted in Figure 13(a) shows that energy consumption proportionately increases with an increasing number of cluster heads in the network because the cluster head plays the role of validating Blocks. Further, cluster

**Table 4**

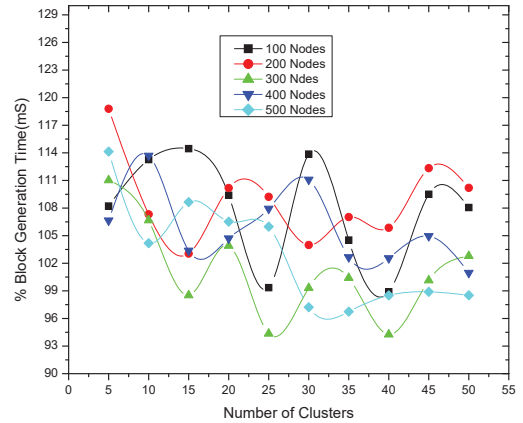
The Parameters for the simulation

Network Area	1000×1000m <sup>2</sup>
Device Radio Range	300m
Fog device CPU capacity	9900MIPS - 83000MIPS(Million instruction per second)
Smartphone CPU capacity	14000 MIPS
Fog device RAM capacity	8 - 16
Fog device Bandwidth	600 - 300Mbps
Smartphone Bandwidth	100-50Mbps
Fog device Power Consumption Rate(per Hour)	140-95W
Fog device Transmission Power Consumption Rate(per Hour)	10W
Smartphone Power Consumption Rate(per Hour)	25-20W
Smartphone Transmission Power Consumption Rate(per Hour)	2
Transaction Size	1024 bytes
Block Size	10× 1024 bytes
Size of the tasks to be migrated	10-5KB/MB
Instruction required to validate Block	10Million
Instruction in a Task	100-50Million

formation using K-means and cluster head selection algorithm consumes power. In contrast, the graph demonstrates an almost similar amount of power consumption regardless of the number of nodes for a particular cluster. This is significant advantage of running consensus mechanism in the hierarchical network.



(a) Clusters vs Energy Consumptions



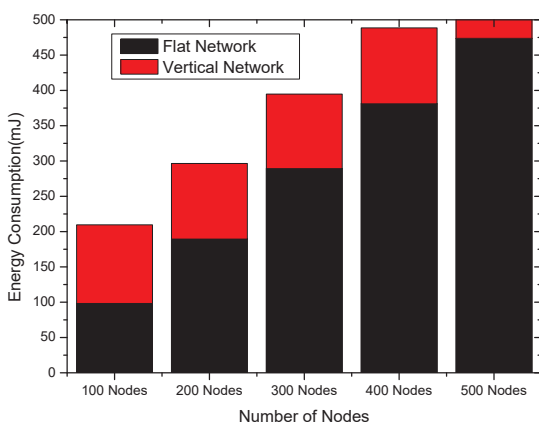
(b) Clusters vs Throughput

**Figure 13:** Performance of modified PoS mechanism in terms of energy consumption and Block generation time

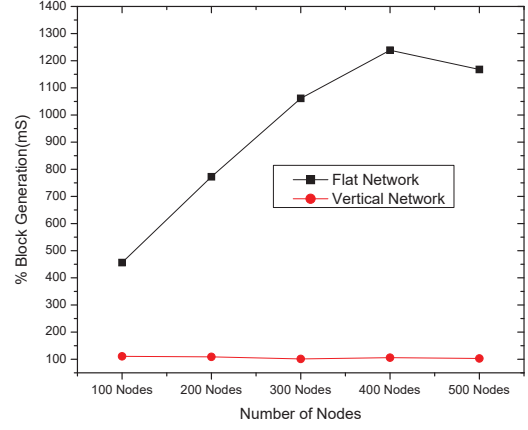
Figure 13(b) shows the Block(100%) generation time for different number of clusters and nodes. The graph depicted in Figure 13(b) shows that Block generation time with a higher number of clusters does not follow a consistently lower or higher trend. Cluster heads gather transactions and make Blocks; so higher number of Blocks are generated per second with a higher number of clusters. On the other hand, higher Block generation time was also found for some higher numbers of the cluster due to delay in verifying Blocks. This indicates that a standard number of cluster needs to be determined to have better outcomes. The ideal number of clusters vary depending on the number of nodes. For instance, the ideal number of cluster for 300 nodes is 25 but it is 35 for 500 nodes.

The performance of the modified PoS is compared with the standard PoS in terms of energy consumption and Block

generation time. In Figure 14(a), the modified PoS shows a significant reduction of energy consumption compared with the standard PoS. In the modified PoS, few selected healthy Miners validate a Block, but the standard PoS requires more than 50% node participates in the Block validation process, which results in higher energy consumption. Energy consumption of modified remains almost constant for a particular number of clusters with a higher number of nodes, whereas energy consumption of PoS keeps increasing when the number of nodes in the network is increased.



(a) Nodes vs Energy consumption



(b) Nodes vs Block generation time

**Figure 14:** The comparison of performance between modified PoS and standard PoS in terms of energy consumption and throughput

The Block generation time of modified PoS and standard PoS is demonstrated in Figure 14(b). The graph depicted in Figure 14(b) shows that the Block generation time standard PoS is higher than modified PoS. In standard PoS, different nodes transmit their transactions to one leader node and confirmed Block is needed to broadcast throughout the network for validation. Consequently, the process consumes higher energy, as depicted in Figure 14(a) and requires a longer time for broadcasting the Block throughout the network. Further, the modified PoS selects some healthy Miners based on reputation, performance and stake, but standard PoS nominates a Miner based only on investment or stake. Therefore, Block generation time is lower in the cluster-based network with modified PoS.

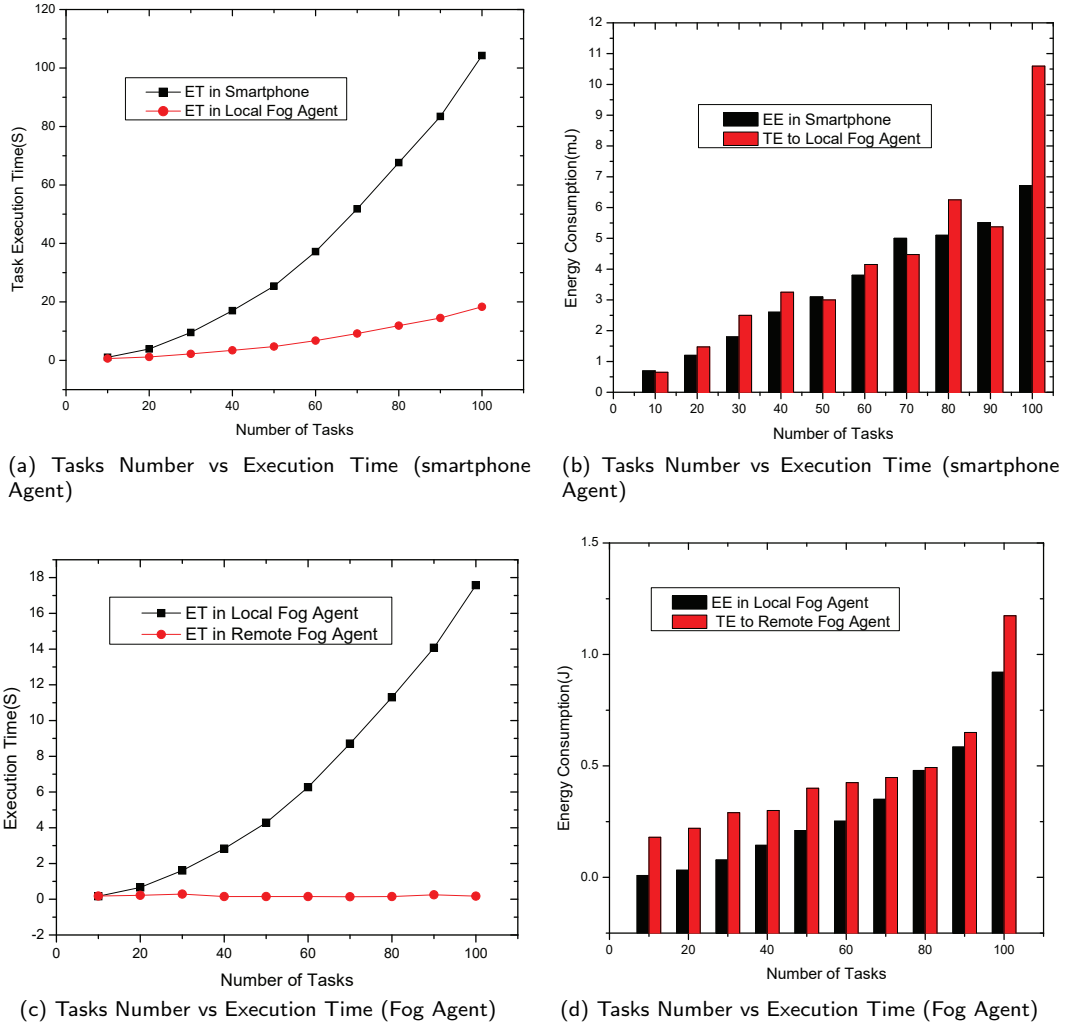
**Task Migration Algorithm:** The Blockchain leveraged task migration approach was also executed in the above mentioned simulated network. The Hungarian assignment algorithm was implemented using Java Programming. We analysed the performance of the task migration methods with respect to variable numbers of tasks such as 10, 20, 30, ..., 100. The task migration algorithm was run by smartphone Agent and Fog Agent. The smartphone transmits tasks to the local Fog Agent. The local Fog Agent utilizes Blockchain to transfer the tasks to other neighbouring or remote Fog Agents. Every Agent applies FCFS( First come First Service) as CPU scheduling to process their jobs. The performance of the task migration method is discussed in terms of the following two metrics.

- **Energy Consumption:** This refers to the energy required for a local Agent to locally execute a task or transmit the task to a foreign/remote Fog Agent.
- **Execution Time:** This refers to the time required for a local Agent to execute a task locally or receive a response for the task from a remote Agent.

EE, ET, TE and TT in figure 15 and 16 are acronyms of Execution Energy, Execution Time, Transmission Energy and Transmission Time, respectively.

The graph for a variable number of tasks vs execution time and a variable number of tasks vs energy consumption are depicted in figure 15 for the smartphone Agent and local Fog Agent when they execute a set of heavyweight tasks(5-10 MB). The processing capabilities of a local Fog Agent are higher than that of a smartphone Agent. As a result, the graph in figure 15(a) shows that the completion time of every set of tasks required in the local Fog Agent is less than that of completion time in the smartphone Agent. The smartphone demonstrated longer queue delay than local Fog





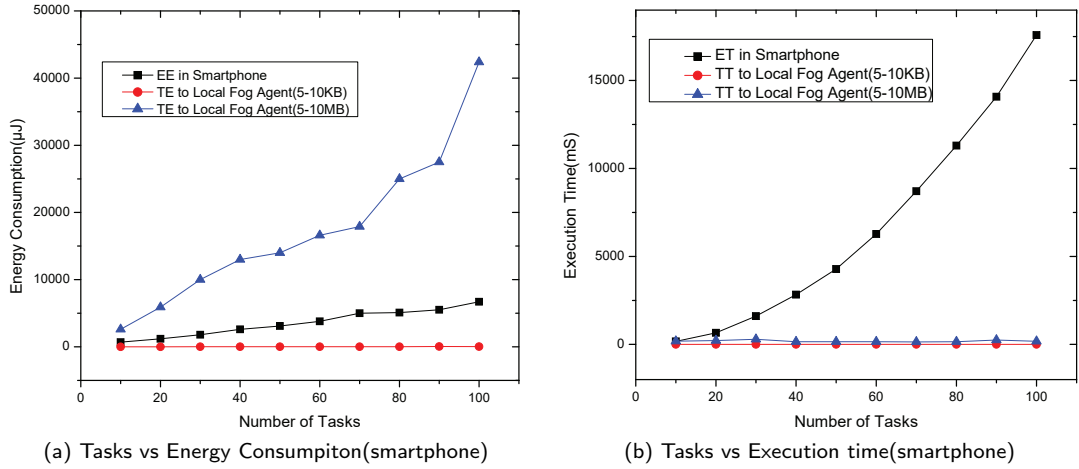
**Figure 15:** The response time and energy consumption for local execution and transmission

Agent in the simulation environment. On the other hand, the graph in figure 15(b) shows that smartphone consumes more energy or hardly a bit less energy (for the number of tasks 50 and 70) to transmit heavyweight tasks to the local Fog Agent than to execute those locally. The graph shows that the smartphone cannot save energy if the tasks are offloaded to the local Fog Agent.

Figure 15(c) demonstrated that if the local Fog Agent outsourced tasks to neighbouring or remote Agents, the overall response time for those tasks is reduced. The foreign Fog Agents parallel execute the assigned tasks. On the other hand, figure 15(d) depicts that the local Fog Agent had to spend higher energy to transmit tasks to foreign Agents than to execute those tasks locally. Consequently, tasks should be divided as delay sensitive or energy sensitive.

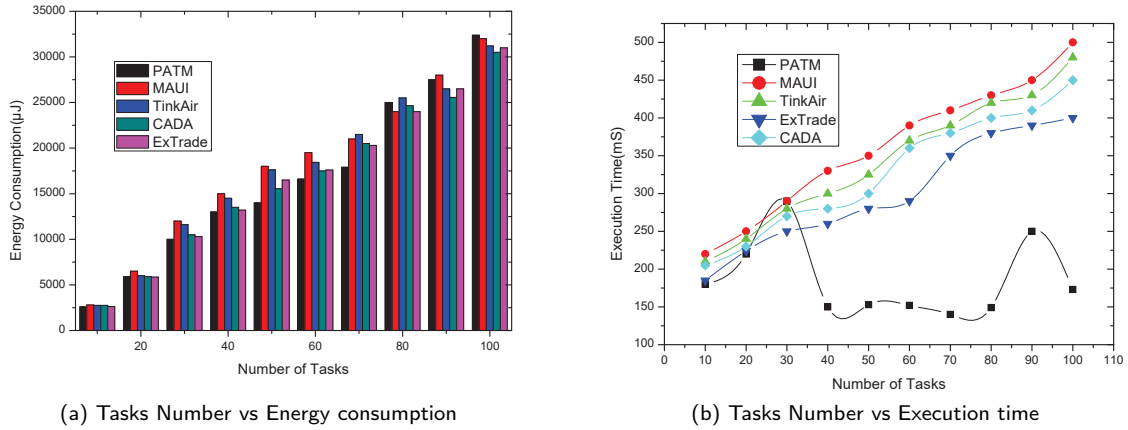
The energy consumption and time for data transmission to remote devices depend on the task's size. The smartphone transmits two kinds of tasks: lightweight(5-10KB) and heavyweight tasks(5-10MB) to the embedded Fog Agents. The effect of task's data size on execution time and energy consumption is shown in figure 16(a) and 16(b). Migrating lightweight tasks needs less energy consumption than that of heavyweight tasks. The smartphone benefits lower transmission energy consumption and time if the transmitted task's size is small.

The energy consumption of five offloading approaches is depicted in Figure 17(a). This energy consumption includes the energy required for a task's transmission and execution. The proposed Patient Agent-based task migration(PATM)improves energy consumption over other methods when the number of tasks is not many. The PATM consumed high energy for the larger number of tasks because the Hungarian assignment algorithm costs much in



**Figure 16:** The comparison of performance for lightweight and heavyweight tasks

terms of energy and time for a large number of tasks. Overall, the PATM saves 1.81% and 8.45% energy in comparison to ExTrade and MAUI approaches, respectively.

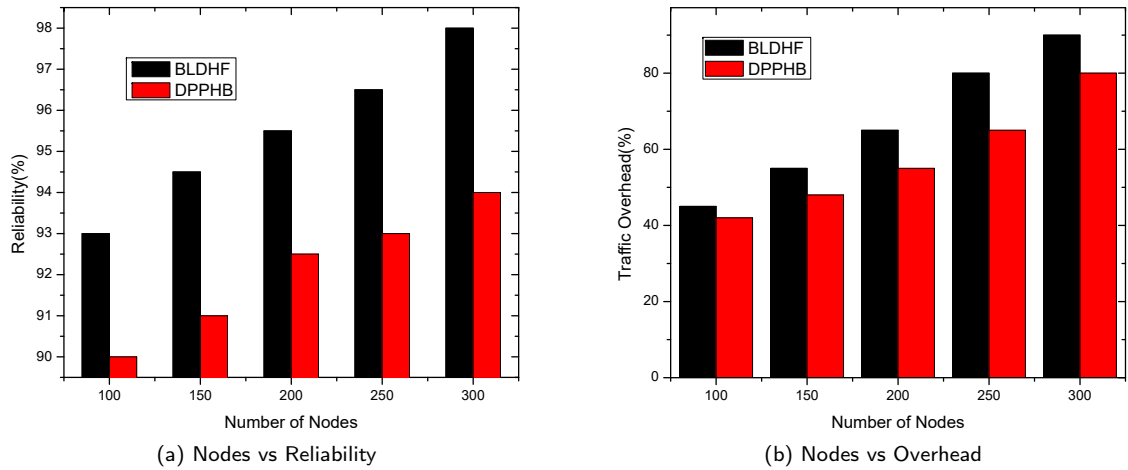


**Figure 17:** The comparison of performance among few offloading approaches

The comparison of the execution time among five offloading approaches is depicted in Figure 17(b). The PATM improves the execution time over other approaches because the Hungarian method chooses some remote Fog devices to optimize the execution of all the tasks. Other migration approaches serially assign a task to a remote Fog device. Other approaches show higher execution time as the number of tasks is increasing, whereas the PATM shows almost constant execution time for the increasing number of tasks. The PATM not only decides to offload but also optimally assigns tasks to remote Fog Agents. Overall, the proposed tasks assignment improves execution time 38.28% over the ExTrade approach that shows the lowest execution time among the existing methods.

## 5. Security Analysis

The code for the ring signature and secret key sharing module are downloaded from [73] and [77], respectively. We compared our eHealth architecture (BLDHF) with an existing eHealth architecture (DPPHB) with respect to reliability and communication overhead. These two performance metrics are related to the security protocol. The graph depicted in figure 18(a) shows that our eHealth achieved higher reliability than DPPHB (A Decentralized Privacy-Preserving Healthcare Blockchain for IoT) because of our decentralized key management and multiple instances of the Patient Agent at three layers. On the other hand, the graph depicted in figure 18(b) displayed that the security mechanism in our eHealth caused higher communication overhead than DPPHB. In our system, an Agent requires to collect a certain number of segments of the data encryption from other neighbouring Agents to form the complete secret key. This method causes communication overhead while authenticating and exchanging secret keys.



**Figure 18:** The comparison of performance between two eHealth architectures

We also used Scyther[78], a formal methods tool to verify the authentication process. Scyther measured the strength of the authentication protocol in our architecture against security attack. Figure 19 shows the outcome of our authentication protocol in Scyther. The automated process of these tools provides checking for authentication, secrecy and message integrity. Scyther analyses the performance of security protocol regarding the following parameters.

1. **Alive:** Scyther can test the aliveness of the communication parties so that they can perform events successfully and be available at any time. This indicates the analysis of a DoS attack.
2. **Secret:** Secret means that the role is secret and there are no attacks within bound, data will require a parameter term to verify the claim. The user needs to pre-set the parameters before testing the claim. This proves that the proposed protocol is protected and ensures the confidentiality of data is provided.
3. **Nisynch:** It is a No-injective Synchronization. This term ensures the successful synchronization, no reply attack, and mutual authentication. This term is used to check if the security protocol safeguards against the replay attack.
4. **Niagree:** The integrity of data can be verified by using the non-injective agreement on message. The term ensures that the original data from the legitimate source is not modified over the communication channel.

Figure 19 shows **OK** for the above claims, which indicates that the applied authentication protocol can withstand different kinds of security attacks.

The architecture needs to be discussed in terms of basic security requirements: Confidentiality, Integrity and Availability.

1. **Confidentiality:** Like Cloud, heterogeneous Fog devices with diverse security methods or no security are deployed by different stakeholders. Sensing and processing of health record by Fog devices is susceptible to malicious attack. In our architecture, the same Patient Agent for a patient replicated in the Smartphone, Fog device and Cloud can safeguard health record from malicious attack. The sensitive medical data is analyzed in the homogeneous replicated Patient Agent to preserve patient's privacy or confidentiality. Further, patient's

Scyther results: verify

Claim	Status	Comments
DPA_RPM I DPA_RPM,I1 Nisynch	Ok Verified	No attacks.
DPA_RPM,I2 Niagree	Ok Verified	No attacks.
DPA_RPM,I3 Secret kir	Ok Verified	No attacks.
DPA_RPM,I4 Secret k(I,R)	Ok Verified	No attacks.
R DPA_RPM,R1 Nisynch	Ok Verified	No attacks.
DPA_RPM,R2 Niagree	Ok Verified	No attacks.
DPA_RPM,R3 Secret kir	Ok Verified	No attacks.
DPA_RPM,R4 Secret k(I,R)	Ok Verified	No attacks.

**Figure 19:** The attack result from Scyther tools

record stored in Blockchain decentralized ledger distributed among multiple public servers encounters the potential privacy leakage because patient has to provide the server with his or her private/public key to decrypt the ciphertext while retrieving health data. This retrieval results in a potential privacy leakage[43]. The replicated Patient Agent can protect the patient's privacy while retrieving health data from the public domain if an individual Agent dedicated to a Patient for creating and managing keys is installed in the public server. Further, a patient's identity is completely anonymous in the Blockchain because of using a ring signature.

2. Integrity: The Fog Agent stores Block's header in the fashion of linked list, which ensures health data integrity.
3. Availability: There is multiple Patient Agent at different levels to process health data and multiple Cloud server to store health data that facilitates the access of health data from multiple points. Therefore, the architecture is providing the users with high availability.

Furthermore, the security strength of the proposed decentralized eHealth architecture is discussed in terms of some attacks, including DoS, mining, storage, and dropping attack[1]. The attack description and mitigation are illustrated here.

1. Dropping Attack: A Dropping Attack occurs when a cluster head drops the transactions. But this is unlikely to happen because the cluster head will lose its reputation and share once it is identified as malicious. If the cluster head is down or malicious and cluster members do not receive transactions for verification, the consensus mechanism should select another node as the cluster head. The cluster member temporarily stores transactions until the Block containing the transaction is confirmed in the Blockchain. Therefore, lost transactions can be retrieved from the cluster members.
2. Storage Attack: A group of malicious nodes can store and corrupt the Blockchain ledger and make health record inaccessible to intended parties. The Blockchain ledger is stored in the Cloud server. Patient Agents for different users use different Cloud servers. Many Cloud server contains the exact copy of the complete Blockchain ledger. The Fog devices also store a chain comprising Block's header without data, required to prove the integrity of the ledger. Data can be retrieved even if some Cloud servers corrupts the ledger because the headers stored in Fog can be used to reinstate corrupt Blocks. This makes a Storage Attack unlikely to be successful.
3. Mining Attack: A 51% attack is called a mining attack where more than 50% nodes can control the network. We divide the entire network into clusters, and the cluster head is responsible for collecting transactions from that cluster members. The cluster head is changed depending on the performance, reputation and locked share after a certain period. A super leader is randomly chosen from the cluster heads. So, nodes from a particular region cannot collude for a mining attack.
4. Denial of Service attack: Denial of Service attack means to shut down the usual activities of a machine through flooding unwanted traffic, causing the legitimate user unable to access the machine. In our eHealth architecture,

the Patient Agent is replicated at three different levels. The Patient Agent executing in the smartphone resumes the services through replicating a Patient Agent at another device at Fog or Cloud level when the Patient Agent is under the DoS attack. Further, a node needs to lock its coin to participate in mining. A user needs to pay a transaction fee to include the transactions in the Blockchain. The locked coin and transaction fee safeguard the system from a DoS attack by the registered nodes.

5. **Selfish Mining:** Selfish Miners attempts to increase their share by not broadcasting mined blocks throughout the network for some period and then releases several Blocks at a time making other Miners lose their Blocks. With our PoS, a super leader can make a certain number of Blocks. In every round, the new super leader is randomly selected to organize transactions into a Block. This approach can reduce the possibilities of such an attack.

The Table 5 presents the comparative analysis of our architecture with other existing Blockchain based frameworks.

**Table 5**

The comparative analysis of the our eHealth system with existing systems

Comparison Criteria	Proposed eHealth system	Existing system -1[33]	Existing system-2[16]
Fault tolerance	High, multiple instances of a PA at three layers manage health data	High for the Blockchain Multi-access Edge Network but low for the sensor network	Medium, single agent controls and manage Blockchain
Confidentiality, Integrity and Availability(CIA)	High CIA, homogeneous PA processes sensitive medical data using ring signature, Edge nodes maintain Blockchain for metadata, multiple PAs ensures service availability	low confidentiality due to third parties' involvement in processing health data, integrity High because of using Blockchain, availability limited due to centralized broker	confidentiality is medium, integrity is high, availability is low due to centralized Blockchain controller
Cyber Attacks	Withstand Ransomware, and DoS attacks	Local processing unit and a universal broker are vulnerable to Ransomware and DoS attack	centralized PA is vulnerable to many cyberattacks
Data Immutability	Yes	Yes	Yes
Secure and energy efficient migration	A privacy aware Blockchain leveraged task migration method	No such approach was designed	No such approach was designed
Interoperability	Yes	Yes	Yes
Scalability	Medium, many resources are required	High	High
Service Reliability	High	Medium	Medium
Consensus Mechanism	Lightweight consensus mechanism was proposed	Existing Proof of Work(PoW) consensus mechanism was used, high computational cost	Modified PoW, medium level of computational cost
Communication Overhead	High traffic due to exchange security keys	Low because security management module was not included	Medium, limited exchanges of security key

## 6. Conclusions

In this paper, we designed an eHealth system that deployed multiple instances of a software Patient Agent at three layers: sensing, NEAR processing and FAR processing layer. This makes the eHealth system more reliable and fault-tolerant. We also described how the PA could be adopted on the 5G architecture. The dedicated Patient Agent software can manage the resources of 5G network slices to embrace the Blockchain technologies in processing health data. The eHealth system includes a modified Blockchain PoS consensus mechanism and a privacy-aware task offloading algorithm. In this eHealth architecture, homogeneous Patient Agents( instances of the same Patient Agent) uses digital ring signature and SSS(Shamir's Secret Sharing) to ensure a secure communication channel between them. The performance analysis demonstrated that the proposed eHealth system could perform the processing of health data in near real-time using Blockchain technologies. The adoption of Blockchain technologies in healthcare is challenged with a massive amount of health data continuously streamed from wearable sensors. Not all medical data generated from continuous patient monitoring does not need to be stored with the same security and privacy level. Health data can be disseminated among multiple health repositories(EHR, EMR, PHR and Blockchain EHR) in accordance with patient's privacy preferences. Our future work is to develop a dynamic storage selection algorithm soliciting patient's preferences regarding his or her privacy and security.

## References

- [1] Ashutosh Dhar Dwivedi, Gautam Srivastava, Shalini Dhar, and Rajani Singh. A decentralized privacy-preserving healthcare blockchain for iot. *Sensors*, 19(2):326, 2019.
- [2] Bin Cao, Yixin Li, Lei Zhang, Long Zhang, Shahid Mumtaz, Zhenyu Zhou, and Mugen Peng. When internet of things meets blockchain: Challenges in distributed consensus. *IEEE Network*, 2019.
- [3] Robab Abdolkhani, Kathleen Gray, Ann Borda, and Ruth DeSouza. Patient-generated health data management and quality challenges in remote patient monitoring. *JAMIA Open*, 2019.
- [4] Maxim Chernyshev, Sherali Zeadally, and Zubair Baig. Healthcare data breaches: Implications for digital forensic readiness. *Journal of medical systems*, 43(1):7, 2019.
- [5] Stojan Kitanov and Toni Janevski. Introduction to fog computing. *The Rise of Fog Computing in the Digital Era*, page 1, 2018.
- [6] Peng Zhang, Jules White, Douglas C Schmidt, Gunther Lenz, and S Trent Rosenbloom. Fhircchain: applying blockchain to securely and scalably share clinical data. *Computational and structural biotechnology journal*, 16:267–278, 2018.
- [7] Jun Zou et al. *Accountability in Cloud Services*. PhD thesis, Macquarie University, Faculty of Science and Engineering, Department of     , 2016.
- [8] Mohammad Aazam and Eui-Nam Huh. Dynamic resource provisioning through fog micro datacenter. In *Pervasive Computing and Communication Workshops (PerCom Workshops)*, 2015 *IEEE International Conference on*, pages 105–110. IEEE, 2015.
- [9] Enzo Baccarelli, Paola G Vinueza Naranjo, Michele Scarpiniti, Mohammad Shojafar, and Jemal H Abawajy. Fog of everything: Energy-efficient networked computing architectures, research challenges, and a case study. *IEEE access*, 5:9882–9910, 2017.
- [10] Sunitha Safavat, Naveen Naik Sapavath Naveen, and Danda B Rawat. Recent advances in mobile edge computing and content caching. *Digital Communications and Networks*, 2019.
- [11] Rima Gibbings and Nilmini Wickramasinghe. A systematic framework to assess emrs and ehrs. In *Theories to Inform Superior Health Informatics Research and Practice*, pages 403–413. Springer, 2018.
- [12] Tsipi Heart, Ofir Ben-Assuli, and Itamar Shabtai. A review of phr, emr and ehr integration: A more personalized healthcare and public health policy. *Health Policy and Technology*, 6(1):20–25, 2017.
- [13] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008.
- [14] George Drosatos and Eleni Kaldoudi. Blockchain applications in the biomedical domain: A scoping review. *Computational and Structural Biotechnology Journal*, 2019.
- [15] Md Ashraf Uddin, Andrew Stranieri, Iqbal Gondal, and Venki Balasubramanian. Continuous patient monitoring with a patient centric agent: A block architecture. *IEEE Access*, 6:32700–32726, 2018.
- [16] Md Ashraf Uddin, Andrew Stranieri, Iqbal Gondal, and Balasubramanian. A patient agent to manage blockchains for remote patient monitoring. *Studies in health technology and informatics*, 254:105–115, 2018.
- [17] Paul Dunphy and Fabien AP Petitcolas. A first look at identity management schemes on the blockchain. *IEEE Security & Privacy*, 16(4):20–29, 2018.
- [18] Laure A Linn and Martha B Koo. Blockchain for health data and its potential use in health it and health care related research. In *ONC/NIST Use of Blockchain for Healthcare and Research Workshop*. Gaithersburg, Maryland, United States: ONC/NIST, pages 1–10, 2016.
- [19] Ariel Ekblaw, Asaph Azaria, John D Halamka, and Andrew Lippman. A case study for blockchain in healthcare:     medrec     prototype for electronic health records and medical research data. In *Proceedings of IEEE open & big data conference*, volume 13, page 13, 2016.
- [20] Huawei Zhao, Peidong Bai, Yun Peng, and Ruzhi Xu. Efficient key management scheme for health blockchain. *CAAI Transactions on Intelligence Technology*, 3(2):114–118, 2018.
- [21] Ao Lei, Haitam Cruickshank, Yue Cao, Philip Asuquo, Chibueze P Anyigor Ogah, and Zhili Sun. Blockchain-based dynamic key management for heterogeneous intelligent transportation systems. *IEEE Internet of Things Journal*, 4(6):1832–1843, 2017.
- [22] Shreshth Tuli, Redowan Mahmud, Shikhar Tuli, and Rajkumar Buyya. Fogbus: A blockchain-based lightweight framework for edge and fog computing. *arXiv preprint arXiv:1811.11978*, 2018.



- [23] Venkata N Inukollu, Taeghyun Kang, and Nina Sakhnini. Design constraints and challenges behind fault tolerance systems in a mobile application framework. In *2015 10th International Design & Test Symposium (IDT)*, pages 159–160. IEEE, 2015.
- [24] NGMN Alliance. Description of network slicing concept. *NGMN 5G P*, 1, 2016.
- [25] Jose Ordóñez-Lucena, Pablo Ameigeiras, Diego Lopez, Juan J Ramos-Munoz, Javier Lorca, and Jesus Folgueira. Network slicing for 5g with sdn/nfv: Concepts, architectures, and challenges. *IEEE Communications Magazine*, 55(5):80–87, 2017.
- [26] Ismaila Adeniyi Kamil and Sunday Oyinlola Ogundoyin. Lightweight privacy-preserving power injection and communication over vehicular networks and 5g smart grid slice with provable security. *Internet of Things*, 8:100116, 2019.
- [27] Min Chen, Yongfeng Qian, Yixue Hao, Yong Li, and Jeungeun Song. Data-driven computing and caching in 5g networks: Architecture and delay analysis. *IEEE Wireless Communications*, 25(1):70–75, 2018.
- [28] Redowan Mahmud, Fernando Luiz Koch, and Rajkumar Buyya. Cloud-fog interoperability in iot-enabled healthcare solutions. 2018.
- [29] Amir M Rahmani, Tuan Nguyen Gia, Behailu Negash, Arman Anzanpour, Iman Azimi, Mingzhe Jiang, and Pasi Liljeberg. Exploiting smart e-health gateways at the edge of healthcare internet-of-things: a fog computing approach. *Future Generation Computer Systems*, 78:641–658, 2018.
- [30] Prabal Verma and Sandeep K Sood. Cloud-centric iot based disease diagnosis healthcare framework. *Journal of Parallel and Distributed Computing*, 116:27–38, 2018.
- [31] Chrystinne Oliveira Fernandes and Carlos José Pereira De Lucena. A software framework for remote patient monitoring by using multi-agent systems support. *JMIR medical informatics*, 5(1), 2017.
- [32] Dong Yuan, Jiong Jin, John Grundy, and Yun Yang. A framework for convergence of cloud services and internet of things. In *Computer Supported Cooperative Work in Design (CSCWD), 2015 IEEE 19th International Conference on*, pages 349–354. IEEE, 2015.
- [33] MD Abdur Rahman, M Shamim Hossain, George Loukas, Elham Hassanain, Syed Sadiqur Rahman, Mohammed F Alhamid, and Mohsen Guizani. Blockchain-based mobile edge computing framework for secure therapy applications. *IEEE Access*, 6:72469–72478, 2018.
- [34] Kristen N Griggs, Olya Ossipova, Christopher P Kohlios, Alessandro N Baccarini, Emily A Howson, and Thair Hayajneh. Healthcare blockchain system using smart contracts for secure automated remote patient monitoring. *Journal of medical systems*, 42(7):130, 2018.
- [35] Yi Chen, Shuai Ding, Zheng Xu, Handong Zheng, and Shanlin Yang. Blockchain-based medical records secure storage and medical service framework. *Journal of medical systems*, 43(1):5, 2019.
- [36] X. Liang, J. Zhao, S. Shetty, J. Liu, and D. Li. Integrating blockchain for data sharing and collaboration in mobile healthcare applications. In *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pages 1–5, Oct 2017.
- [37] James Brogan, Immanuel Baskaran, and Navin Ramachandran. Authenticating health activity data using distributed ledger technologies. *Computational and Structural Biotechnology Journal*, 16:257–266, 2018.
- [38] William J Gordon and Christian Catalini. Blockchain technology for healthcare: facilitating the transition to patient-driven interoperability. *Computational and structural biotechnology journal*, 16:224–230, 2018.
- [39] Tharuka Rupasinghe, Frada Burstein, Carsten Rudolph, and Steven Strange. Towards a blockchain based fall prediction model for aged care. In *Proceedings of the Australasian Computer Science Week Multiconference*, page 32. ACM, 2019.
- [40] Ali Dorri, Salil S Kanhere, and Raja Jurdak. Towards an optimized blockchain for iot. pages 173–178. ACM, 2017.
- [41] Edoardo Gaetani, Leonardo Aniello, Roberto Baldoni, Federico Lombardi, Andrea Margheri, and Vladimiro Sassone. Blockchain-based database to ensure data integrity in cloud computing environments. 2017.
- [42] Oscar Novo. Blockchain meets iot: An architecture for scalable access management in iot. *IEEE Internet of Things Journal*, 5(2):1184–1195, 2018.
- [43] Md Mehedi Hassan Onik, Satyabrata Aich, Jinhong Yang, Chul-Soo Kim, and Hee-Cheol Kim. Blockchain in healthcare: Challenges and solutions. In *Big Data Analytics for Intelligent Healthcare Management*, pages 197–226. Elsevier, 2019.
- [44] Tsung-Ting Kuo, Hyeon-Eui Kim, and Lucila Ohno-Machado. Blockchain distributed ledger technologies for biomedical and health care applications. *Journal of the American Medical Informatics Association*, 24(6):1211–1220, 2017.
- [45] Rui Guo, Huixian Shi, Qinglan Zhao, and Dong Zheng. Secure attribute-based signature scheme with multiple authorities for blockchain in electronic health records systems. *IEEE Access*, 6:11676–11686, 2018.
- [46] Sobia Yaqoob, Muhammad Murad Khan, Ramzan Talib, Arslan Dawood Butt, Sohaib Saleem, Fatima Arif, and Amna Nadeem. Use of blockchain in healthcare: A systematic literature review. *International Journal of Advanced Computer Science and Applications*, 10(5), 2019.
- [47] Suvini P Amaraweera and Malka N Halgamuge. Internet of things in the healthcare sector: Overview of security and privacy issues. In *Security, Privacy and Trust in the IoT Environment*, pages 153–179. Springer, 2019.
- [48] Luca Faramondi, Gabriele Oliva, Roberto Setola, and Luca Vollero. Iiot in the hospital scenario: Hospital 4.0, blockchain and robust data management. In *Security and Privacy Trends in the Industrial Internet of Things*, pages 271–285. Springer, 2019.
- [49] Shangping Wang, Yinglong Zhang, and Yaling Zhang. A blockchain-based framework for data sharing with fine-grained access control in decentralized storage systems. *IEEE Access*, 6:38437–38450, 2018.
- [50] Min Chen, Yin Zhang, Yong Li, Shiwen Mao, and Victor CM Leung. Emc: Emotion-aware mobile cloud computing in 5g. *IEEE Network*, 29(2):32–38, 2015.
- [51] Kai Lin, Fuzhen Xia, Wenjian Wang, Daxin Tian, and Jeungeun Song. System design for big data application in emotion-aware healthcare. *IEEE Access*, 4:6901–6909, 2016.
- [52] M Shamim Hossain and Ghulam Muhammad. Emotion-aware connected healthcare big data towards 5g. *IEEE Internet of Things Journal*, 5(4):2399–2406, 2017.
- [53] Min Chen, Jun Yang, Jiehan Zhou, Yixue Hao, Jing Zhang, and Chan-Hyun Youn. 5g-smart diabetes: Toward personalized diabetes diagnosis with healthcare big data clouds. *IEEE Communications Magazine*, 56(4):16–23, 2018.
- [54] Pradip Kumar Sharma, Saurabh Singh, Young-Sik Jeong, and Jong Hyuk Park. Distblocknet: A distributed blockchains-based secure sdn architecture for iot networks. *IEEE Communications Magazine*, 55(9):78–85, 2017.
- [55] Fangyu Gai, Baosheng Wang, Wenping Deng, and Wei Peng. Proof of reputation: a reputation-based consensus protocol for peer-to-peer

- network. In *International Conference on Database Systems for Advanced Applications*, pages 666–681. Springer, 2018.
- [56] Kejiao Li, Hui Li, Hanxu Hou, Kedan Li, and Yongle Chen. Proof of vote: A high-performance consensus protocol based on vote mechanism & consortium blockchain. In *2017 IEEE 19th International Conference on High Performance Computing and Communications; IEEE 15th International Conference on Smart City; IEEE 3rd International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, pages 466–473. IEEE, 2017.
- [57] Ittay Eyal, Adem Efe Gencer, Emin Gün Sirer, and Robbert Van Renesse. Bitcoin-ng: A scalable blockchain protocol. In *NSDI*, pages 45–59, 2016.
- [58] Kevin Peterson, Rammohan Deeduvanu, Pradip Kanjamala, and Kelly Boles. A blockchain-based approach to health information exchange networks. In *Proc. NIST Workshop Blockchain Healthcare*, volume 1, pages 1–10, 2016.
- [59] Gideon Greenspan. Multichain private blockchain white paper. [Online]. Available: <http://www.multichain.com/download/MultiChain-White-Paper.pdf>. [Accessed: 19-Sept-2018]., 2015.
- [60] Stephanie B Baker, Wei Xiang, and Ian Atkinson. Internet of things for smart healthcare: Technologies, challenges, and opportunities. *IEEE Access*, 5:26521–26544, 2017.
- [61] Yang Lu. Blockchain: A survey on functions, applications and open issues. *Journal of Industrial Integration and Management*, 3(04):1850015, 2018.
- [62] ABE Cuervo, D-k Cho, A Wolman, S Saroiu, R Chandra, and P Bahl. Making smartphones last longer with code offload. In *8th international conference on Mobile systems, applications, and services*, pages 49–62.
- [63] Byung-Gon Chun, Sunghwan Ihm, Petros Maniatis, Mayur Naik, and Ashwin Patti. Clonecloud: elastic execution between mobile device and cloud. In *Proceedings of the sixth conference on Computer systems*, pages 301–314. ACM, 2011.
- [64] Sokol Kosta, Andrius Aucinas, Pan Hui, Richard Mortier, and Xinwen Zhang. Thinkair: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading. In *2012 Proceedings IEEE Infocom*, pages 945–953. IEEE, 2012.
- [65] Ting-Yi Lin, Ting-An Lin, Cheng-Hsin Hsu, and Chung-Ta King. Context-aware decision engine for mobile cloud offloading. In *2013 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, pages 111–116. IEEE, 2013.
- [66] Mahbub E Khoda, Md Abdur Razzaque, Ahmad Almogren, Mohammad Mehedi Hassan, Atif Alamri, and Abdulhameed Alelaiwi. Efficient computation offloading decision in mobile cloud computing over 5g network. *Mobile Networks and Applications*, 21(5):777–792, 2016.
- [67] Hui Yang and Jonathan M Garibaldi. Automatic detection of protected health information from clinic narratives. *Journal of biomedical informatics*, 58:S30–S38, 2015.
- [68] Prateek Jindal, Dan Roth, and Carl A Gunter. Detecting privacy-sensitive events in medical text. Technical report, 2013.
- [69] David Sánchez, Montserrat Batet, and Alexandre Viejo. Detecting sensitive information from textual documents: an information-theoretic approach. In *International Conference on Modeling Decisions for Artificial Intelligence*, pages 173–184. Springer, 2012.
- [70] Welderufael B Tesfay and Jetzabel Serna-Olvera. Towards user-centered privacy risk detection and quantification framework. In *2016 8th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, pages 1–5. IEEE, 2016.
- [71] Yuanyu Zhang, Shoji Kasahara, Yulong Shen, Xiaohong Jiang, and Jianxiong Wan. Smart contract-based access control for the internet of things. *arXiv preprint arXiv:1802.04410*, 2018.
- [72] Yinghui Luo, Yiqun Chen, Qiang Chen, and Qinglin Liang. A new election algorithm for dpos consensus mechanism in blockchain. In *2018 7th International Conference on Digital Home (ICDH)*, pages 116–120. IEEE, 2018.
- [73] Leijurv. leijurv/java-projects, Mar 2015.
- [74] Avi Kak. Lecture 8: Aes: The advanced encryption standard. *Lecture Notes on Computer and Network Security*, Purdue University, URL: <https://engineering.purdue.edu/kak/compsec/NewLectures/Lecture8.pdf>, 2016.
- [75] Sivanagaswathi Kallam. Diffie-hellman: Key exchange and public key cryptosystems. *Master degree of Science, Math and Computer Science, Department of India State University, USA*, pages 5–6, 2015.
- [76] Harshit Gupta, Amir Vahid Dastjerdi, Soumya K Ghosh, and Rajkumar Buyya. ifogsim: A toolkit for modeling and simulation of resource management techniques in the internet of things, edge and fog computing environments. *Software: Practice and Experience*, 47(9):1275–1296, 2017.
- [77] SpinResearch. Spinresearch/rustysecrets, Aug 2018.
- [78] Cas JF Cremers. The scyther tool: Verification, falsification, and analysis of security protocols. In *International Conference on Computer Aided Verification*, pages 414–418. Springer, 2008.