*Mathematical Biosciences and Engineering*

http://www.aimspress.com/journal/MBE

MBE, 18(6): 7344–7362.
DOI: 10.3934/mbe.2021363
Received: 18 June 2021
Accepted: 19 August 2021
Published: 30 August 2021

*Research article*

# Cost-efficient service selection and execution and blockchain-enabled serverless network for internet of medical things

**Abdullah Lakhan[1,8,*], Mazhar Ali Dootio[1,*], Ali Hassan Sodhro[2,3,7], Sandeep Pirbhulal [4,5], Tor Morten Groenli[6], Muhammad Saddam Khokhar[1] and Lei Wang[7]**

[1] Research Lab of AI and Information Security, Benazir Bhutto Shaheed University Lyari, Karachi, Sindh Pakistan

[2] Department of Computer and System Science, Mid Sweden University, Ostersund, Sweden

[3] Department of Computer Science, Kristianstad University, SE-291 88 Kristianstad, Sweden

[4] Department of Information Security and Communication Technology, Norwegian University of Science and Technology, Gjovik 2815, Norway

[5] Norwegian Computing Center, P.O. Box 114, Blindern, Oslo 0314, Norway

[6] Kristiania University College, Department of Technology, Mobile Technology Lab

[7] Shenzhen Institues of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518000, China

[8] College of Computer Science and Artificial Intelligence, Wenzhou University, 325035, China

* **Correspondence:** abdullahrazalakhan@gmail.com, mazharaliabro@bbsul.edu.pk.

**Abstract:** These days, healthcare applications on the Internet of Medical Things (IoMT) network have been growing to deal with different diseases via different sensors. These healthcare sensors are connecting to the various healthcare fog servers. The hospitals are geographically distributed and offer different services to the patients from any ubiquitous network. However, due to the full offloading of data to the insecure servers, two main challenges exist in the IoMT network. (i) Data security of workflows healthcare applications between different fog healthcare nodes. (ii) The cost-efficient and QoS efficient scheduling of healthcare applications in the IoMT system. This paper devises the Cost-Efficient Service Selection and Execution and Blockchain-Enabled Serverless Network for Internet of Medical Things system. The goal is to choose cost-efficient services and schedule all tasks based on their QoS and minimum execution cost. Simulation results show that the proposed outperform all existing schemes regarding data security, validation by 10%, and cost of application execution by 33% in IoMT.

**Keywords:** serverless; IoMT; workflow; service selection; cost-efficient scheduling

## 1. Introduction

The Internet of Medical Things (IoMT) collects medical equipment and apps that use online computer networks to link to healthcare IT systems. Machine-to-machine communication, which is the foundation of IoMT, is enabled by medical devices outfitted with Wi-Fi. IoMT devices connect to cloud systems like Amazon Web Services, which store and analyse collected data. Healthcare IoT is another name for IoMT [1]. Remote patient monitoring for those with chronic or long-term diseases, tracking patient prescription orders, and the location of patients admitted to hospitals are all examples of IoMT, as are patients' wearable mHealth devices that can provide information to carers. Medical gadgets that can be converted to or implemented as IoMT technology include infusion pumps that connect to analytics dashboards and hospital beds equipped with sensors that measure patients' vital signs. Telemedicine is the technique of employing IoMT equipment to remotely monitor patients in their homes. This type of care eliminates the need for patients to visit a hospital or a doctor's office every time they have a medical query or change their condition. The security of sensitive data that flows through the IoMT, such as protected health information governed and healthcare insurances become a growing concern for healthcare providers. Because many consumer mobile devices are constructed with Near Field Communication (NFC) radio frequency identification (RFID) tags that allow the devices to communicate information with IT systems, there are now more possible uses of IoMT than previously. Medical equipment and supplies can also be fitted with RFID tags so that hospital workers can keep track of the quantity they have on hand [2].

Furthermore, the present healthcare applications concentrate on telemonitoring patients, tracking their mobility and supplying physicians and relatives with early updates in critical circumstances: the Internet of Things (IoT) and healthcare application and embedded sensors in a patient's body. The cloud computing along with its extension fog node offers distributed IoT based services to the healthcare applications. Some of the problems which need to be addressed are secure communication, servicing cost, handling and mobility, latency control and energy-efficient routing. Due to the digital revolution, there has been a staggering amount of unstructured data, such as videos and images generated in the healthcare sector. A virtual and linked ecosystem of clinical devices has been developed in the healthcare sector, which continually sends out unstructured and potentially unsecured data vulnerable to attack [3]. It is necessary to transmit these data on a channel, which may not also be protected. However, an individual's physiological data includes exceptionally personal and confidential details. Security is, therefore, a predominant need for healthcare applications, mainly if IoT devices equipped with sensors or body area networks are used in the solutions [4].

Generally, these paid services offered by different vendors (e.g., Cloud, Amazon, Cloud, Alibaba, and Azure) to run the applications under their Quality of Service (QoS) requirements. Recently, serverless computing is a model for edge computing execution in which the server is run by the provider and dynamically regulates machine resource allocation. Pricing is based on the actual amount of resources utilized by an application, not on the volume units provisioning servers. However, besides the benefit of serverless edge computing to run the IoT applications, there are many challenges to be further investigated. The tradeoff between cheap cost and demand QoS is a conflicting problem during execution. Due to the external services, security of offloaded data of different users has pose challenge. The primary concern is security and privacy in healthcare solutions, mainly when deployed in a virtual computing environment. In these instances, it is essential to ensure both users and service providers.

So, to have a secure and foolproof cloud-based healthcare system, the security of services should be devised for applications. Therefore, secure and cost-efficient task scheduling under edge computing serverless architecture becoming a challenging problem.

This work formulates the cost-efficient scheduling of Internet of Medical Things workflow tasks in function and blockchain-enable distributed network. The study devise the cost-efficient system which consists of following components.

1. To solve the scheduling problem of workflow healthcare tasks with different constraints such as deadline, precedence constraint, and cost the problem become NP-hard problem. The study devises Function-Based Task Scheduling Blockchain-Enable Framework (FTSB), which consists of different schemes: Function Verification, Function Pool Priority Queue Task Scheduling. The goal is to schedule all tasks in a way; the goal of the trade-off between function cost and deadline of the task could be obtained in the system. Algorithm 1 shows the process of healthcare workflow on different functions based on different schemes. The proposed system has the following components to process the application with its requirements.
2. Function Verification: This component verified the function correctness (e.g., worms free and trojan horse free) before becoming the system's part.
3. Function Pool: This method collected the verified functions for particular applications and saved in the pool.
4. Priority Queue: This priority method order all tasks into the deadline topological order of the application.
5. Task Scheduling: This study devises iterative heuristics to schedule all tasks based on their deadline and minimizes the application's costs.
6. Blockchain-Enable Fog Network: The study implemented blockchain-enable distributed fog-cloud network to execution functions and verify their transactions among their communications.

The goal is to minimize execution cost of workflow tasks during scheduling and process in the network. The considered network comprises of different computing nodes such as fog node and cloud node. The fog node are implemented at the edge of network. Whereas, cloud node is located away from users and access via internet. The proposed system implemented docker containers to run the functions of different vendors to execute the IoMT tasks of the workflow applications. Each workflow is a business task, for instance body sensor generating the data and send to the heartbeat tasks and offloaded to run for the further analysis. Whereas, heartbeat function process on the requested inside container and generates results and save and second back to the users.

The remainder of this paper is organized as follows. Section 2 discusses the related work in this field. Section 3 describes the problem description and problem formulation. Section 4 proposed algorithm framework. Section 5 present the simulation results to evaluate the performance of our algorithm. Section 6 concludes summary of the study.

## 2. Related work

These days, Internet of Medical Things (IoMT) usage in the healthcare domain to support the different applications grows daily. The cloud computing services boost the performances of the IoMT network, which consists of healthcare sensors, network technologies and cloud computing technologies to run different healthcare applications. The fog layers have been used to improve IoMT-based

healthcare systems' capabilities to maintain the latency and delay-sensitive application. They have proven their value by offering rapid response time and low latency. However, many efforts have been made to reduce security risk in the distributed fog cloud network.

The existing studies investigated cost-efficient and latency optimal task assignment problems by proposing their serverless edge computing architectures. The study [1] empowering low-latency applications through a serverless edge computing architecture. The [2] dealt with the Economics of "Serverless" Cloud Computing to minimize the application's execution cost. The study [3] has proposed economic and architectural impact based on Serverless computing. The goal was to minimize the resource cost of the applications during the performance. The study [4] has presented a preliminary review of enterprise serverless cloud computing (function-as-a-service) platforms. The goal is to offer cost-efficient services to microservices applications.

The study [5] has suggested the SPEC cloud group's research vision on FaaS and serverless architectures. The main idea was to offer function as a service based on customers with cheap to perform application events efficiently. The studies [6] and [7] recommended serverless computing for container-based architectures and building a chatbot with serverless computing to run the microservice-based IoMT applications. The objective was cost-based resource allocation was taken into consideration. The IoMT based on serverless computing for the mobile edge and transient fault aware application partitioning computational offloading algorithm in microservices-based mobile cloudlet networks architectures presented in [8–10]. None of the literature research considered the secure cost-efficient scheduling for the microservice workflow IoT application in the serverless edge computing the security efficient framework proposed in [11]. The goal is to minimize the risk of offloaded healthcare data in the system. The proposed system exploited the blockchain-enabled network to verify the secure transaction of data on different nodes. The symmetric security-aware services in distributed investigated in [12–16]. The objective is to minimize lateness, security risks and offloading cost of applications. The shared public keys and private keys were verified based on a centralized control system in the network in these studies [17]. The blockchain-Enable distributed for the healthcare body area network suggested. The main goal is to minimize end to end security risks of applications in these studies [18–21]. The studies [22–25] focused on failure aware mechanism in IoMT network for the healthcare applications to minimize the failure and validation risk in the system. These studies devised many solutions based on travelling salesmen problem where mobility of the applications is optimized in the network.

To the best of our knowledge, This work formulates the cost-efficient scheduling of Internet of Medical Things workflow tasks in function and blockchain-enable distributed network which has not been studied yet with the considered constraints. These studies [26–30] are closely related to our work. However, these studies only considered the execution cost of fine-grained and coarse-grained workloads on the single edge nodes in the considered problem. Therefore, this study considered the workflow applications in which, the system can manage the execution of tasks on different nodes which are connected and validated by the blockchain technology. The security and validation methods of the proposed study are different than existing single node validation and security mechanism.

## 3. Problem description

The proposed system, as shown in Figure 1 consists of different healthcare components. The healthcare application consists of workflows that connect various healthcare sensors and offload tasks to the healthcare fog server for further execution. The cloud providers offer healthcare services such as blood-pressure monitoring, heartbeat monitoring, and many healthcare services with their usage memory and execution time available via gateway application programming interface.

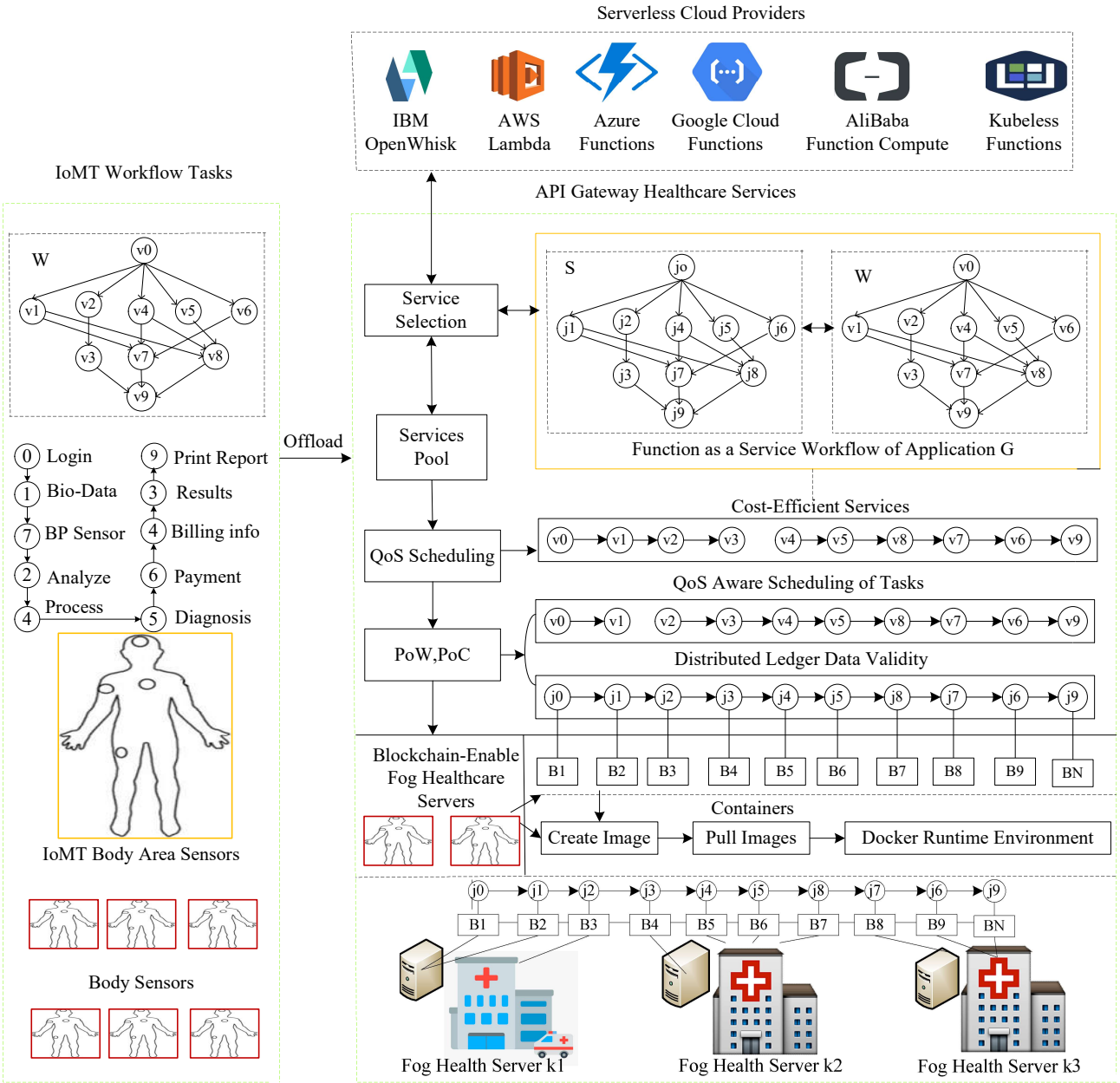



**Figure 1.** Cost-efficient service selection and execution and blockchain-enabled serverless network for internet of medical things.

The service selection is a method in which healthcare services are added to the service pool $S$. The

**Table 1.** Mathematical notation.

| Notation | Description |
|---|---|
| $G$ | IoMT workflow application |
| $V$ | Number of tasks of application $G$ |
| $v_i$ | $i^{th}$ workflow task of application $G$ |
| $v_{id}$ | The deadline of a task $v_i$ |
| $K$ | Number of fog-cloud computing nodes |
| $k$ | The $k^{th}$ computing node of $K$ |
| $\epsilon_k$ | The resource capability of $k^{th}$ node |
| $M$ | Pool of functions |
| $j$ | $j^{th}$ function of node $k$ |
| $C$ | Total number of containers in node $k$ |
| $C_k$ | The $C^{th}$ container of node $k$ |
| $B$ | Number of blocks in the blockchain |
| $B_1$ | The $i^{th}$ block of $B$ |
| $B_{capacity}$ | Capacity of block $B$ |

cloud providers IBM OpenWhisk, AWS Lambda, Azure Functions, Google Cloud, Functions Alibaba, Function Compute, and Kubeless Functions offer healthcare services at different characteristics (e.g., execution cost as per memory). In addition, many fog healthcare servers are connecting (e.g., Each hospital can run additional services for patients). The quality of service (QoS) of workflow applications such as deadline, require assistance to run task data must be optimal in the system. The blockchain-enabled fog healthcare services verify the data authentication based on proof of work (PoW) and proof of creditability (POC) methods in the distributed network. Table 1 defined the description of mathematical notation.

### *3.1. Problem formulation*

The IoMT workflow application is represented by the directed acyclic graph, i.e., $G(V, E)$. For two tasks $v_i, v_z \in V$, an edge $e(v_i, v_z) \in E$ represents the data dependency between task $v_i$ and task $v_z$, which means $v_i$ should complete its execution before $v_z$ starts. The application $G$ has $N$ number of tasks. Where task $v_0$ is the entry task and $v_n$ is the exit task. We imply $data_i$ to denote the original data volume of task $v_i$. Whereas, $data_{i,z}$ denotes generated data volume from task $v_i$ to $v_z$. Each task $v_di$ has deadline inside workflow during process in the system.

The fog-cloud nodes are represented by $\{k = 1, \ldots, K$. Each computing node can create number of containers, i.e., $\{C_1, \ldots, C\}$. Each node is configured with the blockchain consensus blocks, i.e., $\{Block_1, \ldots, Block\}$. The study alias $Block_1$ to $B_1$ to the further process in the study.

The functions pool for tasks of different cloud vendors is represented by $M_i = \{M_i^0, M_i^1, \ldots, M_i^{|M_i|-1}\}$. Whereas $M_i^{jCk}$ is the $j^{th}$ function of node $k$ for $v_i$ which is executing inside container. Whereas, $B_{ijCk}$ is the start time of a task at the $j^{th}$ function in the $k^{th}$ node, and $F_{ijCk}$ is the finish time of the $S_{ijk}$. The execution time of a task is calculated by $T^e_{ijCk}$. The cost of each task is determined in the following way, i.e., $Cost_{ijCk}$ is illustrated by the $S_{ij} = \{T^e_{ij}, Cost_{ijCk}\}$. The binary assignment of each task $v_i$ to the

available function determined as follows.

$$x_{ijCk} = \begin{cases} 1, & S_{ijCk} \text{ function chooses for } v_i \\ 0, & \text{otherwise,} \end{cases} \tag{3.1}$$

Equation (3.1) determines the binary assignment of tasks to the functions.

$$Smart_{data_{i,z}} = \begin{cases} 1, & \sum_{e=1}^{E} Smart_{data_{i,z}} \text{ if tasks data-size equal} \\ 0, & \text{Tempered,} \end{cases} \tag{3.2}$$

$Smart_{data_{i,z}}$ determines the smart-contract rules during communication between tasks and offloading as determined in Eq (3.2). Whereas, $\sum_{e=1}^{E} data_{i,z}$ is the communication of tasks between different thin-client and thick-client. The objective is to reduce cost of workflow tasks under their deadline constraints. The considered problem is formulated as follows.

$$\min Z = \sum_{v_i=1}^{V} \sum_{j=1}^{|M_i|} \sum_{k=1}^{K} \sum_{C=1}^{C} Cost_{ijCk} \times x_{ijCk}. \tag{3.3}$$

$Z$ represented the objective function of the study as defined in Eq (3.3). Subject To,

$$\sum_{j=1}^{|M_i|} \sum_{k=1}^{K} x_{ijCk} = 1, \quad \forall v_i \in V. \tag{3.4}$$

Each task is assigned to only function at any computing node as defined in Eq (3.4).

$$F_{ijCk} = \sum_{j=1}^{|M_i|} \sum_{k=1}^{K} B_{ijCk} + T^{e}_{ijCk} \times x_{ijCk} \leq d_i, \quad \forall v_i \in V, \tag{3.5}$$

The Finish time of tasks must be less their deadlines as defined in Eq (3.5).

## 4. Proposed schemes

To solve the scheduling problem of workflow healthcare tasks with different constraints such as deadline, precedence constraint, and cost the problem become NP-hard problem. The study devises Function-Based Task Scheduling Blockchain-Enable Framework (FTSB), which consists of different schemes: Function Verification, Function Pool Priority Queue Task Scheduling. The goal is to schedule all tasks in a way; the goal of the trade-off between function cost and deadline of the task could be obtained in the system. Algorithm 1 shows the process of healthcare workflow on different functions based on different schemes.

### *4.1. Function verification and pool*

This study devises the function verification method which identify the security requirement of functions before adding to the system pool. The self-replicated and distributed by these systems. The malware must have the capacity to replicate to be labelled as a virus or worm. The following defines these and other malicious software classes.

**Algorithm 1:** FTSB algorithm

**Input :** $\sum_{v_i=1}^{V}$, FaaS preference list $\sum j = 1^M$;
**Output:** $\min Z$

```
begin
    Call SFVM Scheme;
    To verify function add to function pool;
    Call Priority Queue method;
    Sort all tasks into topological order foreach (v = 1 as V) do
        Call Task Scheduling Method;
        Call Blockchain-Enable Scheme;
        Z ← v ← j = {1, 2 . . . , M};
        Obtained Z* Call global searching method;
        if (Z ≤ Z*) then
            Z* ← v ← j = {1, 2 . . . , M};
    Return Z*;
End Main
```

1. Ransomware: is a type of malicious software that, unless a ransom is paid, threatens to publish the victim's data or block access to it permanently. Simultaneously, some basic ransomware can lock the device so that it is not difficult to reverse for a knowledgeable individual and needs a ransom payment to decrypt them.
2. Viruses: A computer virus is a kind of malware that propagates by injecting a copy of itself into another program and being part of it. It spreads from one device to another as it flies, leaving infections. In severity, viruses can range from causing mildly irritating effects to destroying data or software and causing denial-of-service conditions (DoS). Almost all viruses are attached to an executable file, meaning that the virus may reside on a device but will not be transmitted until the malicious host file or program is run or opened by a user. When the host code is executed, it also executes the viral code. The virus infects them. Some viruses, however, overwrite other programs with copies of themselves, mutually deleting the host program. Viruses propagate when the program or document to which they are attached is transferred through the network, a disk, file sharing, or contaminated email attachments from one device to another.
3. Disk Virus Worms are similar to viruses in that they replicate themselves with functional copies and can do the same kind of damage. Worms are standalone applications and do not need a host program or human assistance to propagate, unlike viruses, which require the spread of an infected host file. To spread, the vulnerability to trick users into executing them on the target device or social engineering. Via a loophole in the system, a worm enters a device and takes advantage of file-transport or information-transport features on the system, enabling it to move unaided. More sophisticated worms use encryption, wipers, and ransomware technologies to damage their targets.
4. Memory Virus Another type of malware named after the wooden horse which the Greeks used to infiltrate Troy is a trojan. It is a destructive piece of software that appears legal. Users are usually

fooled by loading it on their systems and executing it. It can conduct any number of attacks on the host after triggered, from annoying the user (popping up windows or changing desktops) to destroying the host (deleting files, stealing data, or activating and spreading other malware, such as viruses). Trojans are also known to build backdoors to give access to the device to malicious users. Trojans, unlike viruses and worms, do not reproduce or self-replicate by infecting other files. Trojans need to be distributed through user activity, such as opening an email attachment or downloading a file from the Internet and running it.

5. Data Virus derives from "robot" and is an automated mechanism that communicates with other network services. Bots also automate tasks and provide data or services that a human being would otherwise perform. Bots are usually used to collect data, such as web crawlers, or automatically communicate with instant messaging (IM), Internet Relay Chat (IRC), or other web interfaces.
6. SOAP: The system can only accept the function which follows the protocol standard development for tasks. SOAP is a specification of the messaging protocol for the sharing of standardized information when integrating web services on computer networks. The goal is to provide extensibility, neutrality, verbosity and autonomy.
7. JSON:Each function should be written in JAVASCRIPT Object Notation (JSON ) form, which is an open standard file format and data exchange format that stores and transmits data objects consisting of attribute-value pairs and array data types using human-readable text. It is a very common data format with a wide variety of uses, such as a replacement for AJAX and XML.
8. Vendors: The study considered the healthcare function of different vendors in order to achieve optimal functions for the tasks.

**Algorithm 2:** SFVM

**Input :** Rules[Trojan, Bot, Worms, Ransomer], $\{j = 1, \ldots, M\}$;
**Output:** $\{j = 1, \ldots, M\}$

```
1 begin
2     foreach (j=1 as M) do
3         Verified;
4         Rules←j;
5         M ← j Added security risk free functions;
6     return M;
```

Algorithm 2 verify each function based on different security rules such as Trojan, Worms, Bot and Ransomer before adding to the function pool.

### *4.2. Queueing priority*

In Priority Queue items, the key value is sorted such that the item with the lowest key value is at the front and the item with the highest key value is at the rear or vice versa. So, based on its key-value, we have given priority to the object. If the value is lower, the higher the priority. The principal methods of a Priority Queue are as follows. The priority queue inserts the item according to its order if an element is inserted into the queue. We're assuming here that high value data has a low priority. The tasks

are sorted in the following order. Front: $v_1, v_2, v_5$. Rear: $v_3, v_4, v_6, v_7, v_9$. All tasks of front queue are scheduled first, and rear queue tasks are scheduled with low priority.

### *4.3. Blockchain mines*

The blockchain mechanism of tasks is defined in the following steps.

- The first block is "Genesis Block" which has the following parameters. (i) $cp_1$ is the computing node to process the mined of a task $s_1$ of $v_1$. Each block has a unique id $b_1$ and previous hash (e.g., The study devises the asymmetric-key cryptography cipher method for encryption that uses a pair of keys, an encryption key, and a decryption key, respectively, called the public key and the private key. This algorithm's key pair consists of a private key that is generated using the same algorithm and a unique public key. It is called Public-Key Cryptography as well).
- A Merkle root is a straightforward mathematical way to verify the Merkle tree data. In cryptocurrency, Merkle roots are used to ensure that data blocks passed between peers on a peer-to-peer network are entire, undamaged, and unaltered.
- Each block for each task has a should contain transaction id.
- The proof of work is implemented to verify the transaction of the block within a network.

#### 4.3.1. Node to node verification

The transaction is verified (validated) by of device in the network against certain validation rules set by the developers of the unique blockchain network. Validated transactions are held in a block and a lock is sealed (hash). The transaction is now part of the blockchain and cannot in any way be changed.

#### 4.3.2. Proof of work

Work proof is a type of zero-knowledge cryptographic evidence in which one party shows to another that for some reason a certain amount of computational effort has been expended. Verifiers may check this investment with minimal effort on their part afterwards.

### *4.4. QoS-efficient scheduling*

The paper introduces a novel service composition method which determines match each Function of different to each task before scheduling. Algorithm 3 takes task preference and function preferences as inputs. Based on the cost and task requirements, the algorithm creates the match list, where each task is to assign to a function which can satisfy its requirements. In the end, it matches all tasks until the list of tasks become empty.

### *4.5. Cost-efficient-rescheduling*

Based on task sequencing, and composition matching list, the scheduler allocates all tasks to functions based on requirements. Algorithm 4 reads the composition list of tasks and functions, then schedule them based on tasks deadline and cost. This process iteratively carry on until tasks are allocated and executed to appropriate functions.

**Algorithm 3:** QoS efficient-scheduling

**Input :** Tasks Preference List $\sum_{G=1}^{N} \sum_{v_i=1}^{V}$, FaaS preference list $\sum j = 1^{M}$;
**Output:** $Match[C_{ij} \times x_{ijk}]$;

**begin**
  **foreach** *(j=1 as M)* **do**
    **foreach** *(v=1 as V)* **do**
    **while** *($S_{ijk} \neq empty$)* **do**
      Search best $S_{ijk}$ is picked for $v_i$;
      Compare each time-shot of service for each task $S_{ijk}$ in $M_i^j$;
      $S_{ijk}$ in $M_i^j$;
    **if** *($S_{ijk}$ in $M_i^j$ is matched)* **then**
      Calculate $C_{ij} \leftarrow x_{ijk}$];
      Add $Match[C_{ij}, x_{ijk}]$;
  $Match[C_{ij}, x_{ijk}]$;
End Main

**Algorithm 4:** Task scheduling

**Input :** $\{\sum_{G=1}^{N}, \sum_{j=1}^{M}, Match[C_{ij}, x_{ijk}]\}$;

**begin**
  **while** *($Match[C_{ij}, x_{ijk}] \neq empty$)* **do**
    **foreach** *($v_i \in G$)* **do**
      Sort all tasks based on task priority methods;
    **foreach** *($j = 1 \in M$)* **do**
      Match each task to each service $Match[C_{ij}, x_{ijk}]$ based on Algorithm 3;
    **if** *($\{B_{ijk} + F_{ijk} \leq vd_i$)* **then**
      $\min Z^* = \sum_{G=1}^{N} \sum_{v_i=1}^{V} \sum_{j=1}^{|M_i|} \sum_{k=1}^{|S_{ij}|} C_{ij} \times x_{ijk}$;
      replace original function $Z \leftarrow Z^*$;
      Final available time slot for all tasks in selected services $M_i^j$ is
       $S_{ij} = \{S_{ij0}, S_{ij0}, \ldots, S_{ij(|S_{ij}|-1)}\}$
      Optimize $Z$;
  Repeat the process until all tasks map to the services;
End Main

## 5. Performance evaluation

The performance evaluation proposed serverless models based on IoMT workflow applications. The simulation environment for the study designed in the Ifogsim with healthcare devices such as Aurdino board as defined in Table 2. All the algorithms designed and implemented in the JAVA language.

**Table 2.** Simulation environment.

| Simulation parameters | Values |
|---|---|
| Simulation Tool | Ifogsim |
| Experimental Machine | Ubuntu X86-64-bit GPU |
| Languages | JAVA, XML, Python |
| Android Phone | Google Nexus 4, 5, and 7S |
| IoMT devices | Aurdino board |

p

**Table 3.** Function of different vendors.

| Providers | FaaS | Cost Dollar per Hour |
|---|---|---|
| IBM OpenWhisk | Linux Amazon GenyMotion | 0.5 |
| AWS Lambda | X86-64-bit AMI | 0.7 |
| Azure Functions | JAVA, XML, Python | 0.3 |
| Google Cloud Functions | Google Nexus 4, 5 and 7S | 0.5 |
| AliBaba Function Compute | 160 times | 0.6 |
| Kubeless Functions | 12 hours | 0.3 |

Table 3 shows the cost of functions of different vendors. To be honest, each of the functions was deployed using a Python 3 runtime with 256 MB of memory. The first benchmark function generated was a factorial function which calculates the result returning factorial 100 fifty times.

### *5.1. System implementation*

The function as a service based serverless system is designed to evaluate the performances of the system based on different workloads as shown in Figure 3. The implemented components of the system is shown in Figure 2. This system which already designed and defined in the our previous published study [6].

#### 5.1.1. IoMT sensors

The Heartbeat Sensor is an electronic system used to measure heart rate, i.e., heartbeat velocity. Body temperature control, heart rate and blood pressure are the basic things we do to keep us safe. We use thermometers and a sphygmomanometer to monitor arterial pressure or blood pressure to calculate body temperature. It is possible to track the heart rate in two ways: one way is to manually check the pulse of the wrists or neck and the other way is to use a Heartbeat Sensor. We have developed a Heart Rate Monitor Device using Arduino and Heartbeat Sensor in this project. The Heartbeat Sensor Concept, the Heartbeat Sensor and the Arduino-based Heart Rate Monitoring Device can be identified using a functional heartbeat sensor. For athletes and patients, controlling heart rate is very important as it determines the state of the heart (just heart rate). There are many methods to calculate the heart rate, and electrocardiography is the most reliable. But using the Pulse Sensor is the best way to track the heart rate. It comes in various shapes and sizes and offers an immediate way to calculate the pulse. Wrist Watches (Smart Watches), Smart Phones, chest belts, etc. are available with heartbeat sensors.

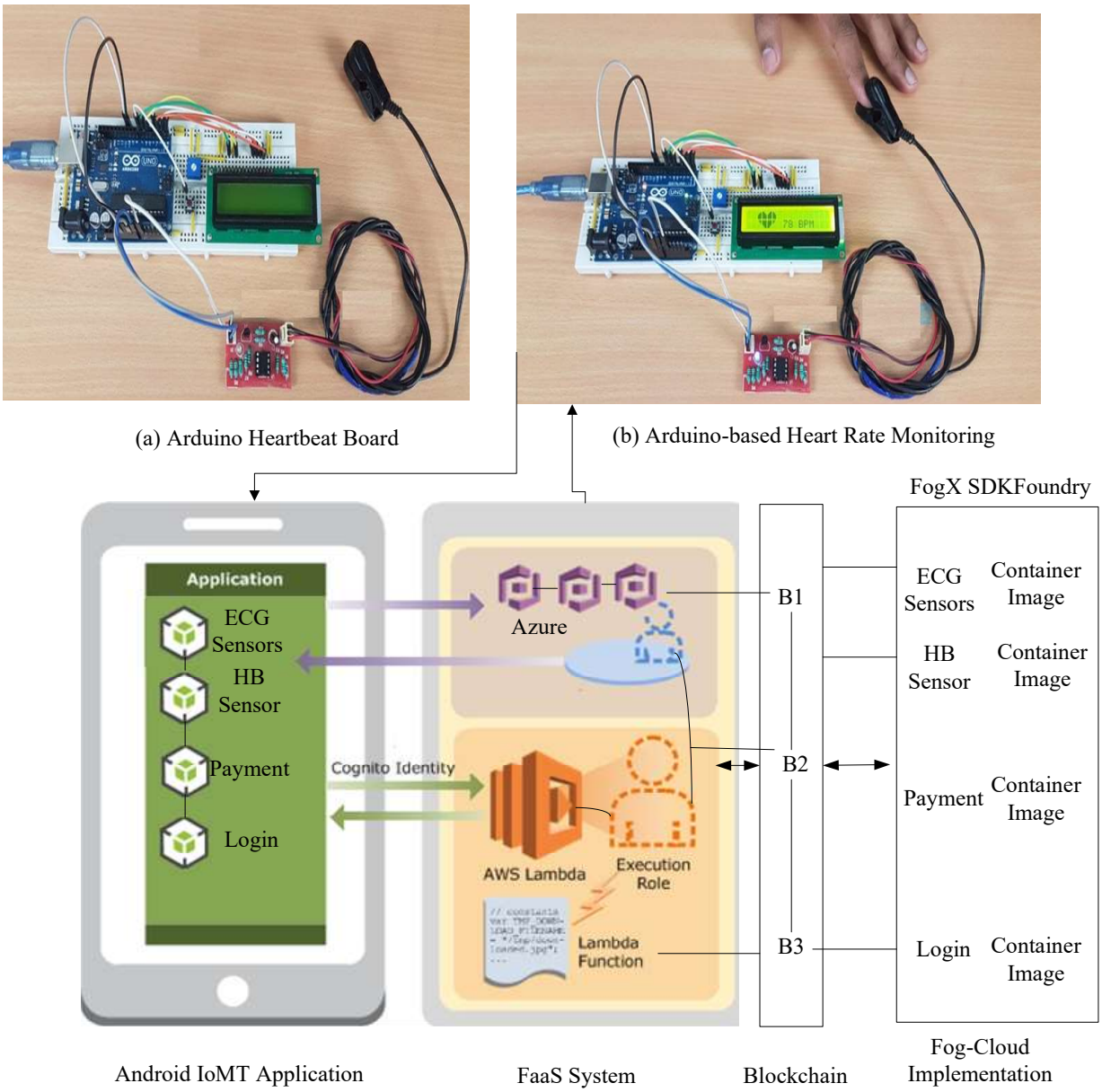



**Figure 2.** Blockchain-enable serverless system.

The heartbeat is measured in beats per minute or in bpm, representing the amount of times in a minute that the heart contracts or expands.

#### 5.1.2. IoMT application

We designed the android IoMT application which consists of four types of different sub-applications such as Cancer aware monitoring, Heartbeat, ECG and EEG monitoring. These applications consisted of workflow tasks as shown in Figure 2, and require different functions to run them. All sensors are connected with an android mobile phone. Whereas, the mobile phone connected to the proposed system which offers services based on functions of different vendors and process them inside containers. The EdgeX Foundry is exploited to design the basic infrastructure for the applications.

#### 5.1.3. Edgex foundry

EdgeX Foundry is a Linux Foundation-hosted, vendor-neutral open-source platform offering a popular mobile framework for IoMT edge computing. There is a series of loosely connected functions of different vendors grouped into different layers inside containers.

### *5.2. Performance metrics*

The study considers the different component calibrations in the serverless model or instance, security, service composition, task sequencing, and scheduling. We measure the performances of IoT workflow application based serverless model via relative percentage deviation (RPD) as follows.

$$RPD(\%) = \frac{Z^* - Z}{Z^*} \times 100\%. \tag{5.1}$$

$Z^*$ displays optimal obtained objective of the study during scheduling. Whereas, $Z$ is the objective function of the study which determines the operational cost of the application.

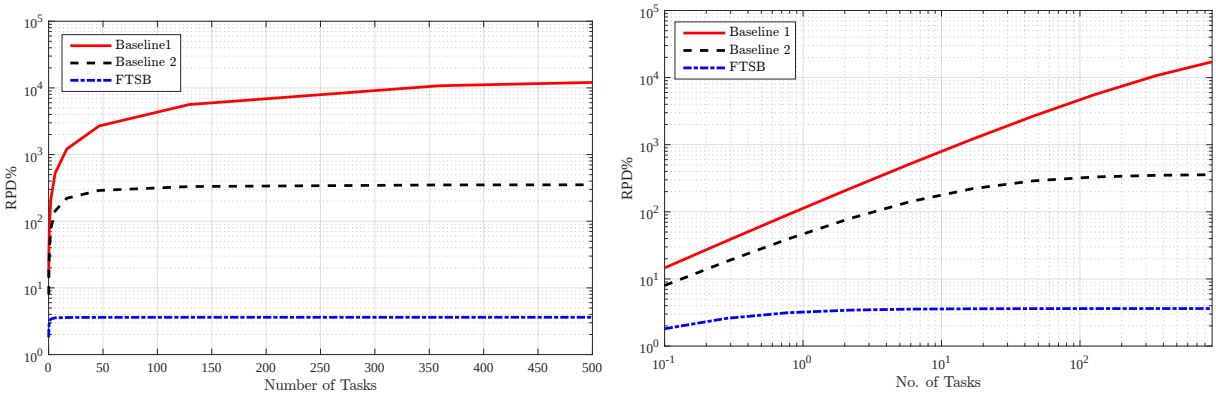
### *5.3. Result discussion*

This part discusses the obtained results of the proposed system and its approaches which are compared to baseline approaches to solve the problem.

#### 5.3.1. Blockchain and function verification

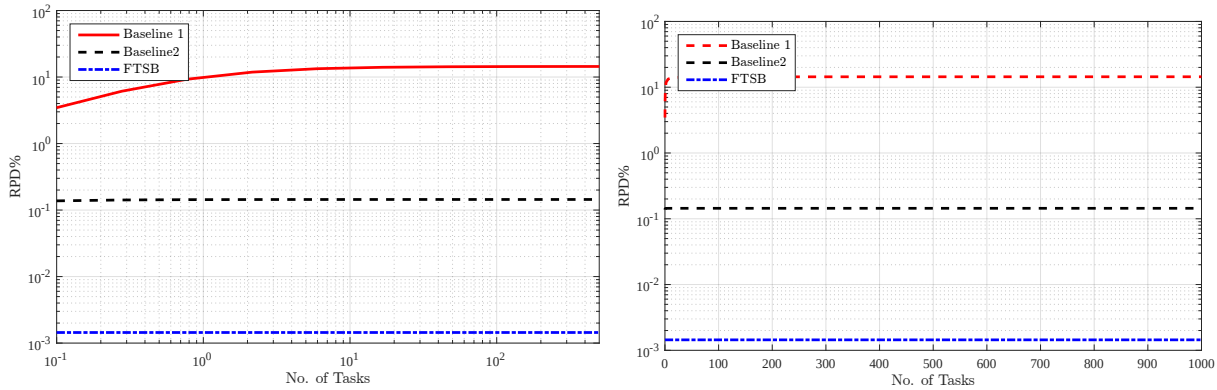
The study suggested the function method to identify the function standard, which should be cleared from any security issue types due to many issues. The existing viruses are Trojan, Worms, and Bot can affect and violated the data of user application. In the study, the system accepts any function in the pool and its verification via different standards. Existing studies such as baseline 1 [1–6] and baseline 2 [1, 5, 7, 9, 11] suggested cost-efficient scheduling systems by exploiting blockchain technologies. The baseline system exploited different for medical care applications based on the blockchain-enabled network. However, they did not focus on function validation and verification before adding to the pool.

Figure 3 (a),(b) shows that, the RDP% performance 500 and 1000 number of workflow healthcare tasks by exploiting proposed FTSB which is outperform all existing system in terms of verification of functions and blockchain-enable perform of healthcare tasks. The main reason behind that, due to




**(a)** Function verification cost **(b)** Blockchain-mine cost

**Figure 3.** Function verification and blockchain-cost.



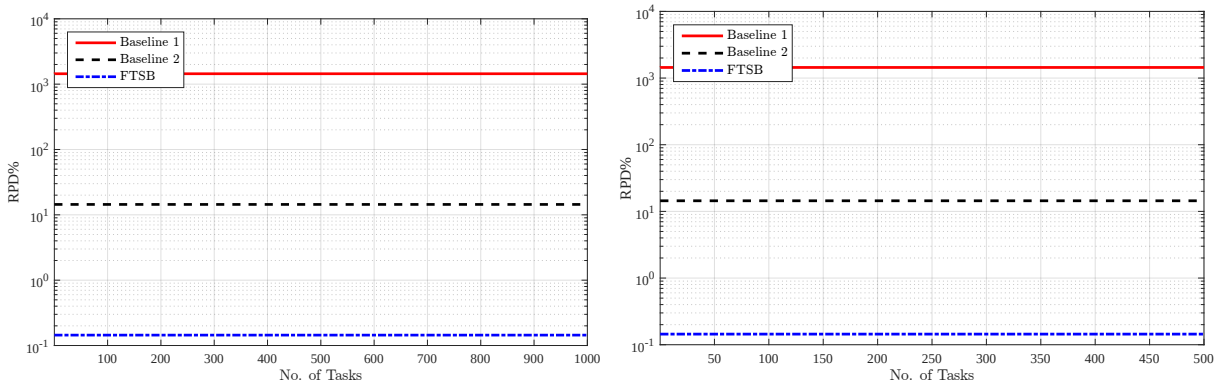

**(a)** Execution-Cost **(b)** Deadline

**Figure 4.** Q0S-scheduling.

immature and effected functions violated the data security and increases the cost of requested tasks during execution. Baseline 1 exploits the static resource provisioning cost model which is always costly in scheduling during variation. Therefore, baseline 1 has a more extended cost than dynamic resource provisioning which is exploited by baseline 2. However, still, the proposed FTSB outperforms all existing approaches in terms of cost.

### 5.3.2. QoS scheduling

The study considered the deadline of healthcare tasks during scheduling in the system. All tasks have different priority during offloading and scheduling in the system. All tasks are sequenced into their precedence constraints in the function aware fog cloud network. All existing studies such as baseline 1 [1–6] and baseline 2 [1, 5, 7, 9, 11] devised genetic algorithm based and dynamic programming to solve the task scheduling problem of workflow healthcare tasks based on resource-provisioning methods (e.g., on-demand, on-reserve, and spot-instances). These studies exploited virtual machines and microservices-based resources which charge based on a pay-as-you-go model. However, they did not focus on the trade-off between resource cost and deadline of tasks during scheduling.




**(a)** Initial cost of functions **(b)** After 30 seconds interval the cost of functions

**Figure 5.** Cost-efficient-scheduling.

Figure 4 (a),(b) shows the performance based on 500, and 1000 number of workflow healthcare tasks by exploiting FTSB approach obtained the optimal results of both execution cost and deadline of healthcare tasks during scheduling on different functions. The main reason is to keep the trade-off between execution cost and deadline with different requirements. Another reason, functions charge for their executions (execution × memory) and different from the existing resource-provisioning model. Therefore, the proposed framework and its schemes FTSB outperform all existing systems and their methods to run healthcare tasks with different requirements. Baseline 1 exploits the static resource provisioning cost model which is always costly in scheduling during variation. Therefore, baseline 1 has a more extended cost than dynamic resource provisioning which is exploited by baseline 2. However, still, the proposed FTSB outperforms all existing approaches in terms of cost.

### 5.3.3. Cost-efficient-rescheduling

Initially, the system selected the plan for all tasks in advance. However, we run the tasks into sequence order, i.e., from the start task to the end. The study does not run tasks into parallel order. Therefore, rescheduling means that a task's initial selection function could be changed with another available function and have small execution cost. All tasks are scheduled one by one due to their precedence constraints requirements. In the function pool, the pool manager checks the existing functions' prices and new available functions in different time-interval for the requested tasks. The pool manager monitors the cost of function after 30 seconds to avoid the overhead of searching in the system.

Figures 5 and 4(a),(b) shows the performance based on 500, and 1000 number of workflow healthcare tasks by exploiting FTSB approach obtained the optimal results of both execution cost and deadline of healthcare tasks during scheduling on different functions. Figure 5 (a),(b) shows that cost-efficient rescheduling for all tasks in different-time slots outperform all existing studies in terms of function cost. However, existing studies selected resources in advance, the runtime variation of resource cost widely ignored in their methods. Baseline 1 exploits the static resource provisioning cost model which is always costly in scheduling during variation. Therefore, baseline 1 has a more extended cost than dynamic resource provisioning which is exploited by baseline 2. However, still, the proposed FTSB outperforms all existing approaches in terms of cost.

## 6. Conclusions and future work

This paper devised the Cost-Efficient Service Selection and Execution and Blockchain-Enabled Serverless system for the Internet of Medical Things. The primary goal is to minimize the services cost in the distributed network when the different healthcare applications run the computing nodes. And another goal is to minimize the security risk of considered function during implementation in the system pool. The blockchain aware mechanism implemented in the distributed service network is based on serverless technology where applications only paid for the execution instead of renting costs for some duration. The simulation results in the discussion part showed that the proposed FTSB Algorithm outperformed and minimized the security and execution of applications. The serverless based resources are more effective than traditional resource assignments in the IoMT, as we achieved the optimal results in the result part.

In the future work, we will extend the IoMT system with additional constraints such as mobility of services, failure of resources and energy consumption of devices during resource researching in the system. The serverless system has an overhead issue. Therefore, serverless can minimize execution costs. However, there is a lot of energy consumption and failure of resource issues in the current version IoMT system. Therefore, after execution of applications on serverless, the results transferring to the cloud-based on security and mobility will be suggested in the existing IoMT system for further improvement for the healthcare applications.

## Acknowledgment

This work was developed and implemented at the research lab of Artificial Intelligence and Information Security in the Department of Computer Science and Technology, Benazir Bhutto Shaheed University, Lyari, Karachi, Sindh, Pakistan. The work is totally testbed and implemented in the practice of Pakistan Hospitals.

This work is financially supported by the Research grant of PIFI 2020 (2020VBC0002), China.

## References

1. L. A. Mastoi, Q. U. Ain, M. Elhoseny, M. S. Memon, M. A. Mohammed, Deep neural network-based application partitioning and scheduling for hospitals and medical enterprises using iot assisted mobile fog cloud, *Enterp. Inf. Syst.*, (2021), 1–23.
2. T. Huang, L. Lan, X. Fang, P. An, J. Min, F. Wang, Promises and challenges of big data computing in health sciences, *Big Data Res.*, **2** (2015), 2–11.
3. A. Lakhan, M. Ahmad, M. Bilal, A. Jolfaei, R. M. Mehmood, Mobility aware blockchain enabled offloading and scheduling in vehicular fog cloud computing, *IEEE Trans. Intell. Transp. Syst.*, 2021.
4. T. Lynn, P. Rosati, A. Lejeune, V. Emeakaroha, A preliminary review of enterprise serverless cloud computing (function-as-a-service) platforms, in *2017 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, (2017), 162–169.

5. A. Lakhan, M. S. Memon, M. Elhoseny, M. A. Mohammed, M. Qabulio, M. Abdel-Basset, et al., Cost-efficient mobility offloading and task scheduling for microservices iovt applications in container-based fog cloud network, *Cluster Comput.*, (2021), 1–23.
6. A. Lakhan, M. A. Mohammed, A. N. Rashid, S. Kadry, T. Panityakul, K. H. Abdulkareem, et al., Smart-contract aware ethereum and client-fog-cloud healthcare system, *Sensors*, **21** (2021), 4093.
7. A. Lakhan, M. A. Dootio, T. M. Groenli, A. H. Sodhro, M. S. Khokhar, Multi-layer latency aware workload assignment of e-transport iot applications in mobile sensors cloudlet cloud networks, *Electronics*, **10** (2021), 1719.
8. M. Hussain, L. F. Wei, A. Lakhan, S. Wali, S. Ali, A. Hussain, Energy and performance-efficient task scheduling in heterogeneous virtualized cloud computing, *Sustainable Comput.: Inf. Syst,*, **30** (2021), 100517.
9. A. Lakhan, X. Li, Transient fault aware application partitioning computational offloading algorithm in microservices based mobile cloudlet networks, *Computing*, **102** (2020), 105–139.
10. A. Lakhan, L. Xiaoping, Energy aware dynamic workflow application partitioning and task scheduling in heterogeneous mobile cloud network, in *2018 International Conference on Cloud Computing, Big Data and Blockchain (ICCBB)*, **2018** (2018), 1–8.
11. A. Lakhan, X. Li, Content aware task scheduling framework for mobile workflow applications in heterogeneous mobile-edge-cloud paradigms: Catsa framework, in *2019 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)*, (2019), 242–249.
12. A. Lakhan1, X. Li, Mobility and fault aware adaptive task offloading in heterogeneous mobile cloud environments, *EAI Endorsed Trans. Mobile Commun. Appl.*, **5** (2019), 16.
13. J. Yun, Y. Goh, J. M. Chung, Dqn based optimization framework for secure sharded blockchain systems, *IEEE Int.Things J.*, 2020.
14. F. Zhang, M. M. Wang, Stochastic congestion game for load balancing in mobile edge computing, *IEEE Int. Things J.*, 2020.
15. A. Lakhan, Q. U. A. Mastoi, M. A. Dootio, F. Alqahtani, I. R. Alzahrani, F. Baothman, et al., Hybrid workload enabled and secure healthcare monitoring sensing framework in distributed fog-cloud network, *Electronics*, **10** (2021), 1974.
16. F. H. Khoso, A. Lakhan, A. A. Arain, M. A. Soomro, S. Z. Nizamani, A microservice-based system for industrial internet of things in fog-cloud assisted network, *Eng. Technol. Appl. Sci. Res.*, **11** (2021), 7029–7032.
17. F. H. Khoso, A. A. Arain, A. Lakhan, A. Kehar, S. Z. Nizamani, Proposing a novel iot framework by identifying security and privacy issues in fog cloud services network, *Int. J.*, **9** (2021), 592–596.
18. A. Lakhan, R. Singh, Implementation of etl tool for data warehousing for non-hodgkin lymphoma (nhl) cancer in public sector, pakistan, *Int. J.*, **9** (2021), 7.
19. A. Lakhan, F. H. Khoso, A. A. Arain, K. Kanwar, Serverless based functions aware framework for healthcare application, *Int. J.*, **9** (2021), 4.

20. M. Waseem, A. Lakhan, I. A. Jamali, Data security of mobile cloud computing on cloud server, *Open Access Libr. J.*, **3** (2016), 1–11.
21. I. A. Jamali, A. Lakhan, D. Kumar, A. R. Mahessar, Energy efficient task assignment algorithm framework in mo-bile cloud computing, *GSJ*, **6** (2018), 171.
22. A. L. Mujeeb-ur Rehman, Z. Hussain, F. H. Khoso, A. A. Arain, Cyber security intelligence and ethereum blockchain technology for e-commerce, *Int. J.*, **9** (2021), 7.
23. A. Lakhan, D. K. Sajnani, M. Tahir, M. Aamir, R. Lodhi, Delay sensitive application partitioning and task scheduling in mobile edge cloud prototyping, in *International Conference on 5G for Ubiquitous Connectivity*, (2018), 59–80.
24. D. K. Sajnani, A. R. Mahesar, A. Lakhan, I. A. Jamali, R. Lodhi, M. Aamir, Latency aware optimal workload assignment in mobile edge cloud offloading network, in *2018 IEEE 4th International Conference on Computer and Communications (ICCC)*, (2018), 658–662.
25. D. K. Sajnani, A. R. Mahesar, A. Lakhan, I. A. Jamali, Latency aware and service delay with task scheduling in mobile edge computing, *Commun. Network*, **10** (2018), 127.
26. A. H. Sodhro, Z. Luo, A. K. Sangaiah, S. W. Baik, Mobile edge computing based qos optimization in medical healthcare applications, *Int. J. Inf. Manage.*, **45** (2019), 308–318.
27. A. H. Sodhro, S. Pirbhulal, V. H. C. De Albuquerque, Artificial intelligence-driven mechanism for edge computing-based industrial applications, *IEEE Trans. Ind. Inf.*, **15** (2019), 4235–4243.
28. M. Muzammal, R. Talat, A. H. Sodhro, S. Pirbhulal, A multi-sensor data fusion enabled ensemble approach for medical data from body sensor networks, *Inf. Fusion*, **53** (2020), 155–164.
29. H. Magsi, A. H. Sodhro, F. A. Chachar, S. A. K. Abro, G. H. Sodhro, S. Pirbhulal, Evolution of 5g in internet of medical things, in *2018 international conference on computing, mathematics and engineering technologies (iCoMET)*, (2018), 1–7.
30. T. Zhang, A. H. Sodhro, Z. Luo, N. Zahid, M. W. Nawaz, S. Pirbhulal, et al., A joint deep learning and internet of medical things driven framework for elderly patients, *IEEE Access*, **8** (2020), 822–832.


AIMS Press