

# Optimal Resource Allocation in Fog Computing for Healthcare Applications

Salman Khan<sup>1,\*</sup>, Ibrar Ali Shah<sup>1</sup>, Nasser Tairan<sup>2</sup>, Habib Shah<sup>2</sup> and Muhammad Faisal Nadeem<sup>3</sup>

<sup>1</sup>Department of Computer Software Engineering, University of Engineering and Technology, Mardan, 23200, Pakistan

<sup>2</sup>Department of Computer Science, College of Computer Science, King Khalid University, Abha, K.S.A

<sup>3</sup>Informatics Complex, H-8, Islamabad, 44000, Pakistan

\*Corresponding Author: Salman Khan. Email: engrsalmanbangash@gmail.com

Received: 31 August 2021; Accepted: 16 November 2021

**Abstract:** In recent years, the significant growth in the Internet of Things (IoT) technology has brought a lot of attention to information and communication industry. Various IoT paradigms like the Internet of Vehicle Things (IoVT) and the Internet of Health Things (IoHT) create massive volumes of data every day which consume a lot of bandwidth and storage. However, to process such large volumes of data, the existing cloud computing platforms offer limited resources due to their distance from IoT devices. Consequently, cloud-computing systems produce intolerable latency problems for latency-sensitive real-time applications. Therefore, a new paradigm called fog computing makes use of computing nodes in the form of mobile devices, which utilize and process the real-time IoT devices data in orders of milliseconds. This paper proposes workload-aware efficient resource allocation and load balancing in the fog-computing environment for the IoHT. The proposed algorithmic framework consists of the following components: task sequencing, dynamic resource allocation, and load balancing. We consider electrocardiography (ECG) sensors for patient's critical tasks to achieve maximum load balancing among fog nodes and to measure the performance of end-to-end delay, energy, network consumption and average throughput. The proposed algorithm has been evaluated using the iFogSim tool, and results with the existing approach have been conducted. The experimental results exhibit that the proposed technique achieves a 45% decrease in delay, 37% reduction in energy consumption, and 25% decrease in network bandwidth consumption compared to the existing studies.

**Keywords:** Fog computing; internet of health things; resource management

## 1 Introduction

With the rapid advancements of smart devices and mobile communication technologies, the Internet of Things (IoT) has become a popular industry. According to an estimate by Cisco, more than 50 billion things have been expected to connect through the Internet by 2020 [1]. This massive growth in IoT devices will continue to increase exponentially in the ongoing decade, and the generated data will



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

be enormously difficult to handle. Among them, the growing attention of the *things* is that the Internet of Health Things (IoHT) is increasing day by day. These things consist of sensors, smart devices, and many other sensors that produce an enormous amount of data to be processed quickly. All these data sent to the server for processing are in the form of tasks and are latency-sensitive. The tasks produce by IoHT devices are dynamic, stochastic, and variant in nature and require an immediate response. Cloud computing is a centralized computed paradigm that can process and store massive data generated by end devices and give them reliable services [2]. However, end devices consume a lot of network bandwidth and burden cloud data centers, which creates communication latency [3]. Consequently, many IoHT delay-sensitive services and applications for healthcare patients cannot be processed and responded quickly by cloud computing. Therefore, this disadvantage of cloud computing has brought an attention and give an emergence of new computing paradigm, called as fog computing [4]. Fog computing has gained significance attention in recent years, and is an effective computing paradigm which renders services to IoT users and its related application domain at the edge of network [5,6].

Fog computing utilizes nodes, such as mobile and IoT devices, by processing the real-time data generated by these devices. Typically, the response time of a fog computing system is very quick, which is quite beneficial for latency-sensitive applications. Fog computing is beneficial for real-time applications; among them, healthcare applications have gained immense popularity in recent years [6]. However, fog computing is still in its infancy, and the devices are not so powerful to execute the request of different IoT devices. Resource management for the incoming request of various IoHT tasks is a challenging issue in fog computing. IoHT applications send patient's critical data and tasks patient's to the fog servers. These critical tasks need to be responded quickly without any delay. storage Fog nodes may forward this data to cloud data centers for processing and storage for long-term analytics. However, fog devices are generally constrained for resources, unlike cloud data centers. The processing and storage capacity of fog computing is limited in nature to fulfill the requirements of diverse and stochastic data [7].

To fulfill the requirements of diverse and dynamic data arriving from end-users to resource constrained devices causes overhead delays and wastage of resources. The resource utilization in fog computing is a challenging issue that must be addressed [6–11]. Proper resource allocation, nodes sorting according to task requirements, and load balancing are critical issues, affecting the fast and timely response desirable for several real-time applications. For instance, in IoHT healthcare critical applications, patients' information and appointment are of great importance to save their lives [4–9]. Hence, considering the criticality of healthcare applications, there is a need to develop an efficient resource utilization of fog resources for diverse applications. This research article proposes a workload-aware dynamic resource allocation and load balancing algorithm for IoHT applications to achieve maximum resource utilization among various fog nodes. This algorithm ameliorates the traditional overhead-based scheduling algorithm for making appropriate resource allocation. The tasks are assigned as per the capacity of processing power. The primary objectives are to minimize the delay, energy, network consumption and maximize average throughput as performance metrics. Individually IoT application consists of fine-grained tasks, but the individual task has the following attributes: CPU response time and workload size.

Thus, dynamic resource allocation and task scheduling in the distributed fog computing with consideration of optimal load balancing are investigated in the paper, which is different from existing works. We are analyzing the following challenges: (i) How to choose a dynamic resource allocation of IoHT tasks for optimal offloading? (ii) How to optimize the load balancing of fog devices during their peak workload? The paper makes the following key contributions:

- The study devises the optimal resource allocation in a fog computing environment for the IoHT delay-sensitive applications. The proposed optimization algorithm consists of task sequencing, dynamic resource allocation, and load balancing. The nodes are parametrized and sorted according to tasks requirements and balances the load to reduce delays and ameliorate the wastage of resources.
- This study considers the IoHT applications that have stringent requirements in terms of their execution and the resources required for their timely completion. Therefore, the study implements node sequencing according to tasks before tasks are assigned to them. The aim is to sort all nodes to allocate resources to tasks and execute them in an optimized way.
- This study adopts dynamic changes in the fog computing environment for IoHT applications, an overhead-based dynamic scheduling algorithm has been suggested. The main goal of the algorithm is to allocate resources to reduce delays, energy, network consumption, execution time, and maximize the throughput.

The rest of this paper is organized as follows; Section 2 highlights the related work; Section 3 presents the fog computing architecture; Section 4 covers the resource allocation and load balancing problem in fog computing; Section 5 presents the proposed resource-scheduling algorithm; Section 6 presents the experimental evaluation and analysis; Section 7 concludes the work highlights future direction.

## 2 Related Work

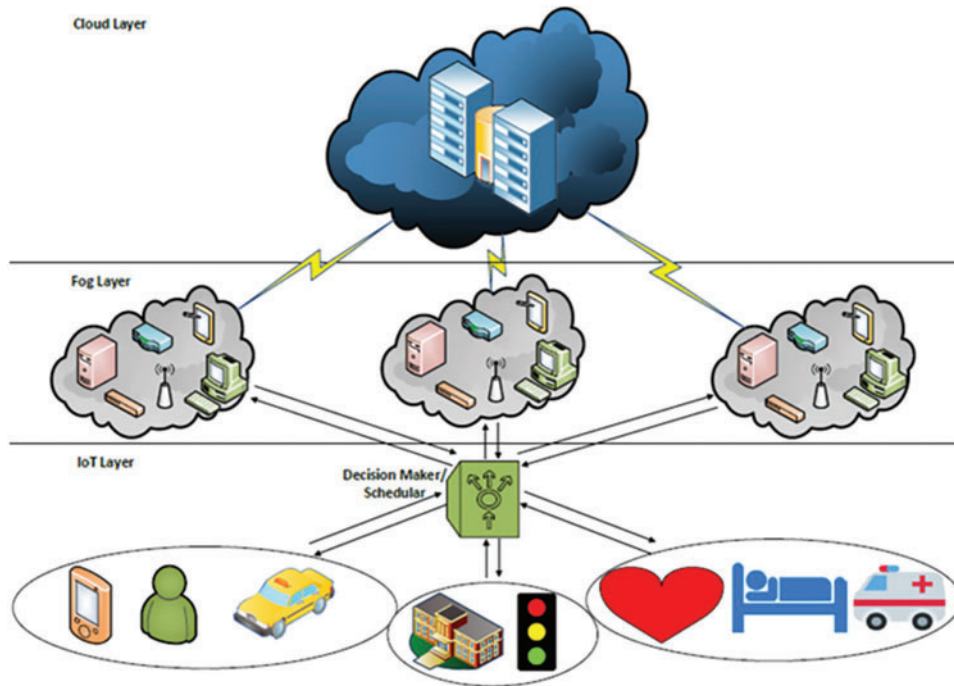
Several existing research works have proposed resource management techniques in fog computing, with specific focus on healthcare system. Resource management is a major concern of fog computing infrastructure designers. The purpose of resource management is to utilize the resource-constrained devices optimally to reduce latency, energy consumption, cost, network usage and provide better quality of service (QoS). Jamil et al. [4] have presented the performance optimization and job scheduling algorithm in fog computing for healthcare critical patients. The purpose of the proposed method is to reduce average delay and energy consumption as an evaluation metrics for delay-sensitive critical healthcare applications. However, they did not consider the dynamic resource allocation and load balancing which raises the problem of overhead delays and wastage of resources. Kumari et al. [6], Ijaz et al. [8], Awaisi et al. [9], Tuli et al. [10] present fog architecture for patient-oriented real-time healthcare applications for data collection, processing and transmission. The authors have highlighted existing issues in healthcare system, such as faulty data and data duplication, data integration, user authentication, data security and privacy. However, they only restricted their work towards healthcare security issues and exploited to mention the dynamic resource allocation and load balancing mechanism for proper resource utilization. Aazam et al. [11], Rahmani et al. [12] presented fog based architecture for emergency alert system and smart e-health gateways to tackle IoT related healthcare system issues. The primary objective of the proposed architecture in fog computing is to overcome the delay, complexity, scalability, mobility, interoperability, and reliability. However, proper resource allocation, and task scheduling are the main concern for healthcare critical applications to overcome delays, energy, network and CPU response time are exploited. Hassan et al. [13] proposed a fog computing-based remote pain monitoring system that collects electromyography (EMG) signals and processes it to detect the pain of different patients. In the proposed model, the information related to pain can be accessed remotely through web application using minimum time and enabled timely medical facilitation to the patients. However, they applied first come first serve (FCFS) scheduling algorithm, which lacks dynamic resource allocation and load balancing to optimize resources and

reduce overhead delays. Lakhan et al. [14], Lakhan et al. [15], Lakhan et al. [16] presented fog-cloud framework for Internet of Vehicle Things (IoVT) and Internet of Medical Things (IoMT). The proposed framework focus on cost-efficient task offloading and task scheduling algorithm in fog-cloud environment. The proposed algorithm framework cost-efficient mobility offloading and task scheduling (CEMOTS), deep neural networks energy cost-efficient partitioning and task scheduling (DNNECTS) and blockchain-enable smart-contract cost-efficient scheduling algorithm framework (BECSAF) consist of task offloading, task sequencing and scheduling. The main objectives are to optimize healthcare application costs and computational costs in the dynamic environment. However, they did not mention other parameters for consideration in healthcare related applications. Mohammad et al. [17], Abdulkareem et al. [18] proposed machine learning approaches for visual anti-spam to detect abnormalities. The proposed approach consists of naïve bayes (NB), random forest (RF), and support vector machine (SVM) classifiers. The primary objectives of the proposed machine learning approaches are to reduce doctors workload, mortality rate, tackling overcrowding during COVID-19 pandemic. However, the proposed approaches is based on multi-natural language and COVID-19 pandemic situation and lacking the fog computing architecture which can be addressed. Ahmad et al. [19], Ahmad et al. [20] proposed multi-device and multi-task orchestration architecture for IoT enterprises. The purpose of the proposed architecture is to provide scalable and flexible operation from tasks generation to mapping and scheduling till allocating them on respective IoT resources in order to reduce round trip time and task dropping rate. The main objective of the optimization is to maximize the flexibility and scalability of the system for newly added devices. However, the proposed architectures mainly lack the concept of load balancing technique for fog-IoT environment. Ahmad et al. [21], Ahmad et al. [22] proposed an adaptive approach for real-time IoT job for formal verification. The proposed approach consider optimal threshold value for task set and provide real time environment for IoT tasks for monitoring and evaluation. The significance of the proposed work is to improve response time, CPU utilization and power consumption. However, the proposed work did not consider resource allocation and load balancing for tasks execution in fog computing. Shaheen et al. [23] proposed framework and present the concept of fog head node to keep track of other nodes in location-awareness and user registration perspective. The primary concern of the work is to reduce latency, network consumption and service time. However, the study did not consider the load balancing for delay-sensitive applications in fog-IoT environment.

To the best of our knowledge, workload-aware dynamic resource allocation and load balancing algorithm for multi-objective optimization under CPU, network constraint has not been studied yet. All the studies mentioned earlier, either solve static resource allocations while ignoring resource utilization or use dynamic resource allocation while missing the load balancing. The overhead of dynamic allocation in resource-constraint devices directly affect real-time applications. Therefore, this study optimizes the utilization of resources for healthcare applications and reduces the overhead delay, energy, network consumption, and execution time and maximizes the average throughput as a performance metric.

### 3 Proposed Fog Computing Architecture

In this section, n-tier fog computing architecture has been described as depicted in Fig. 1. The architecture consists of three layers, namely cloud layer, fog layer, and IoT layer. The proposed architecture is meant for delay-sensitive healthcare applications for efficient utilization of fog computing resources. The illustration of each layer in this architecture has been discussed below.



**Figure 1:** Healthcare system architecture based on fog computing

### 3.1 Cloud Layer

The cloud-computing layer is the topmost layer of the architecture, as shown in Fig. 1. The cloud layer consists of large data centers for massive data processing and storage purposes [24]. It provides services to the end-users in various application domains, such as smart cities, smart transportations, smart health, smart production companies, and virtual reality. All these application areas are delay-sensitive and require a quick response from the data center. However, due to the centralized nature and multi-hop distances from IoT devices [20,21], the issue of latency arises as a main challenging problem in cloud computing for these delay-sensitive applications and cannot meet real-time requirements.

### 3.2 Fog Layer

Fog layer lies in between the cloud layer and IoT layer. The fog layer comprises servers, routers, modems, PC's etc. All the devices with the potential to process, store and communicate data can be considered as fog devices. Fog computing came into existence after the latency issues were faced by the cloud computing systems. Different application domains required faster response and fog computing meets the real time demands from IoT devices [25] and fulfill user satisfaction.

### 3.3 IoT Layer

The IoT layer is the bottom layer of the architecture, as shown in Fig. 1. This layer consists of sensors and actuators to collect real-time data from the surroundings and send it to the fog layer for further processing. There are certain domains in which the IoT layer can play a vital role and improve the quality of service. These application areas include smart transportation, smart healthcare, industrial automation, and emergency response etc. [23–24].

### **3.4 Scheduler**

In this work, we introduce a scheduler or decision-maker module, which is depicted in Fig. 1. This module is significant in fog computing resource management, as it balances the incoming requests or loads equally into fog servers. This technique gives the surety that no overloaded servers will remain in the system. The main advantage of using this technique is to maximize resource utilization, increase throughput, minimize the delay and waiting time and improve overall system reliability and performance.

## **4 Resource Allocation and Load Balancing Problem in Fog Computing**

Resource allocation and load balancing is a persistent problem related to cloud computing and fog computing. If resource allocation is performed in cloud computing, then large data transmission consumes massive bandwidth and causes traffic burden on cloud data centers, and ultimately creates delays for delay-sensitive tasks. On the other hand, resource allocation and load balancing in fog computing are also significant issues because the devices in the fog computing environment are weaker in terms of their processing capability and storage capacity, which need to be addressed by applying appropriate techniques.

This study simulates the workload aware efficient dynamic resource allocation and load balancing problem in fog computing for healthcare applications. The goal is to optimize resource utilization, thereby reducing end-to-end delay, energy, network consumption, and execution time for applications. The goal is also to achieve the average throughput utility by applying the proposed technique.

## **5 Design and Implementation**

Real-time applications are delay-sensitive and need faster response time. Cloud computing systems are centralized and geographically far away from end-users; therefore, it produces communication latency and is unable to address IoT applications' needs. Fog computing emerges as a new computing paradigm and provides essential services to end-user applications to address the issue. However, fog computing resources are limited as compared to cloud computing and the IoT devices send various tasks that are variable in length and stochastic; therefore, to manage the various nondeterministic tasks in the resource-constrained fog devices, various resource management techniques have been applied. This article presents a workload-aware dynamic resource allocation and load balancing technique for healthcare applications to optimize various performance metrics. The proposed approach aims to reduce the delay and energy, network consumption, and maximize the average throughput for critical healthcare applications.

Therefore, there is a need to design and implement a task-scheduling algorithm in fog computing for healthcare applications with the following objectives:

- To efficiently utilize fog resources.
- To minimize the end-to-end delay (Healthcare application).
- To minimize energy consumption, network consumption, and execution time.
- To maximize the average throughput utility.

### **5.1 Case Study**

IoT devices consist of different sensors, actuators, smart gadgets etc., which allow monitoring of different activities for the healthcare system [26]. Fog computing is an emerging technology and is an essential architectural paradigm for Ubiquitous computing. This article presents a case study

for critical healthcare applications in fog computing for different activities. These activities are delay-sensitive and can be performed by fog computing for faster response and more minor delays.

In healthcare, different use case classes have distinct requirements. Some classes are more critical, while some afford to observe delay. Patients require faster response and prompt data analysis to save their lives in a critical or emergency situation. In our healthcare scenario, we use three use cases to find the average delay, which have been elaborated as follows:

- a) Emergency Alert System.** Emergency alert processes the patient's vital data like heartbeat, blood pressure, and blood glucose level obtained from various body sensors. This use case contains patient's critical data to be processed in case of emergency. The data is analyzed and generates timely notifications of patient's health to save their lives.
- b) Patient Appointment System:** This module gets an appointment for patient's, and also ensures the elimination of appointment duplication for the various patients.
- c) Patient Record System:** This use case contains vital information about patient's personal information store in the database. Besides patient's information, it includes doctor information, doctor visit, patient treatment, and lab results. This module also provides for the record of newly admitted patients.

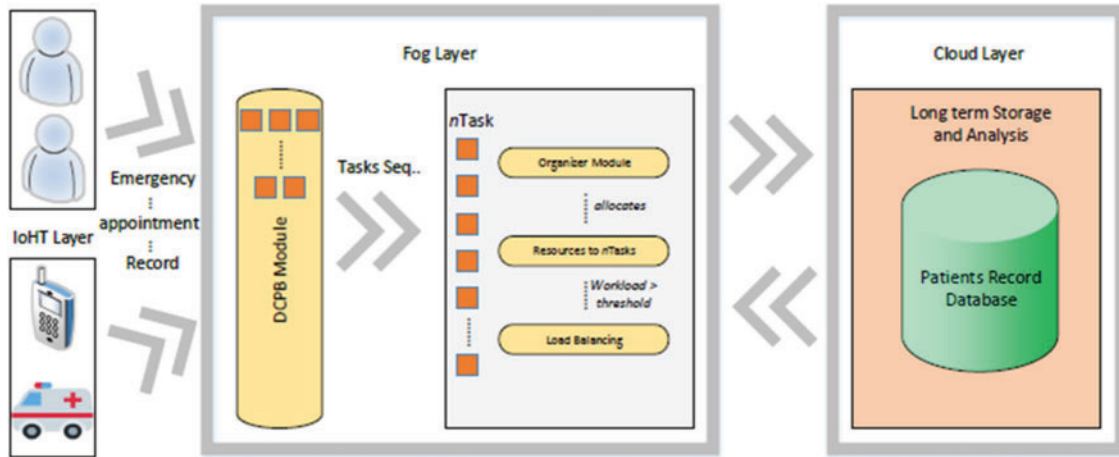
To better recognize this healthcare case study, similar to the work [4], this article uses three application modules in the fog-computing framework.

**A. DCPB:** Data processing and communication board module (DCPB) receives data from the IoHT layer. The tasks are considered critical and non-critical. Critical data such as various sensors send crucial tasks for immediate response and is notified. It receives all the data for the patient's appointment and records but forwards it to the organizer module. The primary concern of this module is to receive all tasks generated from the IoHT layer to sequenced them and forward them further for necessary processing. It also sends back the notification response after processing from an organizing module.

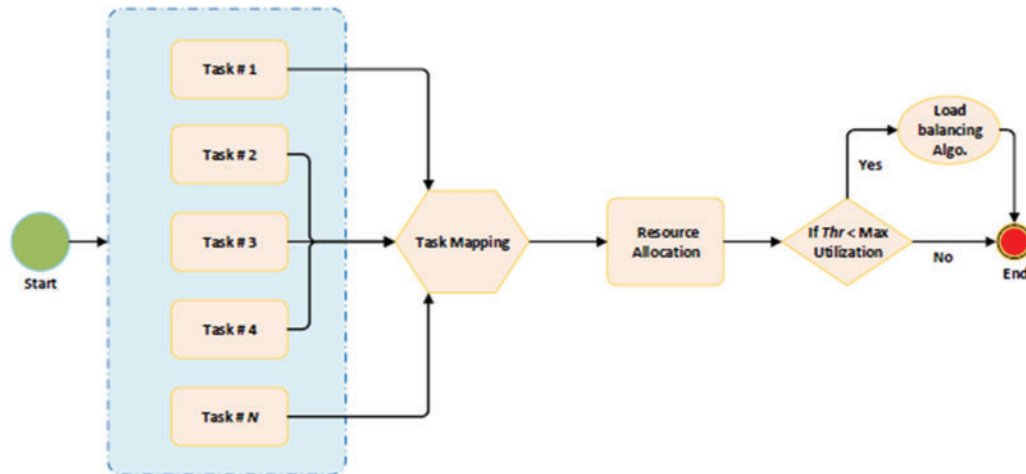
**B. Organizer:** This module acts as a high-level fog device that receives data from the DCPB module. The working nodes are sorted according to the requirements of tasks. The applied algorithm is designed to allocate resources to tasks and check the overhead if any occurs. The fog nodes are continuously checked for CPU overhead; if there found any node that is not capable of executing the request or the node is over-utilized, then the load balancing module will balance the load from over-utilized node to less utilized node, and this process continues until all the tasks are fully executed.

**C. Patients Record Database:** This module receives data from the organizer and sent to the cloud for long-term analytics and storage. The module saves the patients record, which is non-critical, and generates reports sent back to the organizer module about patient's health and hospital visit information. All the modules used in our case study are shown in [Fig. 2](#).

In this paper, we present a systematic approach to control task submission under the optimal threshold to guarantee the successful execution of tasks within their time constraints. The optimal threshold is selected based on the number of fog nodes resources. Moreover, the paper has three main parts: first is the task mapping to appropriate devices, the devices are sorted according to tasks, second the resources are allocated to variant tasks. If there are unscheduled tasks having no response from node resources or some of the resources are underutilized, then we apply a load balancing technique for the optimal utilization of resources in order to reduce the overhead, energy, and response time. [Fig. 3](#) shows the overall flowchart of the proposed approach.



**Figure 2:** Proposed healthcare framework in n-tier computing



**Figure 3:** Flow diagram of the proposed approach

The proposed algorithm dynamically allocates resources to healthcare tasks and balances the load among fog devices to minimize the delay, energy, and network utility. This algorithm searches concurrently in all fog nodes for an available resource  $r$  to serve  $UT_i$  within the required response time. If the start time of  $UT_i$ , when served by  $r$ , is greater than the earliest start time of  $UT_i$ , then the task will be served in its original fog node  $Nd_k$ . Otherwise, the algorithm will shift or migrate  $UT_i$  to other fog node that contains resource  $r$ , and schedule  $UT_i$  to  $r$ . The maximum utilization of the resource  $r$  has been set as 5000 mips threshold value. The pseudocode of the proposed algorithm in this article is given as follows:



**Algorithm 1:** Workload-Aware Dynamic Resource Allocation and Load Balancing Algorithm

---

**Input:** Task list  $T_i$  ( $T_i | i = 1, 2, \dots, n$ ), Node list  $Nd_k$  ( $Nd_k | k = 1, 2, \dots, r$ )  
**Output:** Min [ $D, E, N$ ], Max [ $Tr$ ]  
**Initialization:**  $CU_r$  current utilization of resource  $r$ .  
 $MU_r$  maximum utilization of resource  $r$ .  
 $UT_i^m$  mips required by unscheduled task  
 $R_m$  mips available with the resource  $r$ .  
 $R_t$  is the time available by the resource for  $UT_i^m$   
 $T_i^{est}$  is the earliest starting time by task  $T_i$   
 $R_j$  is the set of fog computing resources in fog nodes  $Nd_k$  that can serve  $UT_i$

```

1  For each ( $T_i \in n$ ) do
2    For each ( $Nd_k \in r$ ) do
3      Sort all nodes  $r$  based on tasks  $n$ 
4      Allocate every  $T_i \in n$  to  $Nd_k \in r$ 
5      If ( $CU_r < MU_r$ ) then
6        If ( $UT_i^m \leq R_m$  and  $R_t \leq T_i^{est}$ ) then
7          Shift  $UT_i$  to  $j$ 
8          Schedule  $UT_i$  to  $R$ 
9          Update  $R_m$ 
10         Update  $R_t$ 
11          $UT_i$  as scheduled
12         Break
13         End If
14       End If
15     End For
16   End For

```

---

When dynamic and stochastic tasks arrive at the decision-maker module, it acts as an intermediate module between fog devices and IoT devices. The decision-maker is also called a scheduler. In this algorithm, we proposed a load balancer, which balances the dynamic load equally to all fog devices and executes them efficiently to reduce the average latency, energy, network consumption and increase average throughput for healthcare applications.

For the implementation of our proposed algorithm, *iFogSim* was chosen because it is an extension of *CloudSim*. *iFogSim* is used for modeling and simulation of cloud-fog computing architecture as well as services and efficient for simulation of resource management techniques for Fog IoT environment.

## 6 Performance Evaluation

We present the performance of our proposed algorithm and evaluate it based on delay, energy consumption, network usage, and throughput.

### 6.1 Experimental Settings

Cloud and fog have different processing capacities and other resources. We assumed that each node has its own processing capacity represented by million instructions per second (MIPS), along with memory and bandwidth usage. In the fog layer, fog nodes could be routers, gateways, workstations, or personal computers with limited processing capacity compared to the Cloud layer, which has servers

or virtual machines in high-performance data centers and is responsible for handling tasks. Therefore, the processing speed of Cloud nodes is much faster than fog nodes. Thus, the characteristics of fog computing in the case of our healthcare scenario are presented in [Tab. 1](#). The parametric values are arbitrary, but we have considered the limitations of a fog environment as compared to a cloud. We implemented the proposed algorithm in *iFogSim* and utilized these parameters to generate our simulation results.

**Table 1:** Value of parameters used for cloud fog-based simulation framework

Parameters	Fog environment	Cloud environment	Unit
Number of nodes	100–500	0–1	node
CPU rate	5600	44800	MIPS
Memory usage	4000	40000	MB
Up bandwidth	10000	100	MB
Down bandwidth	10000	10000	MB
Level	1	0	Level
MIPS	0.0	0.01	Rate
Busy power	107.339	1648	Watt
Idle power	83.4333	1332	Watt

## 6.2 Simulation Setup

The settings of the simulations setup are presented in [Tab 2](#), and the simulation program was developed in Java with Eclipse editor using *iFogSim* [27]. *iFogSim* was chosen because it is has been implemented as an extension of CloudSim, and hence inherits several features from CloudSim.

**Table 2:** Software/Hardware configuration

Software/Hardware	Description
System	Intel <sup>(R)</sup> Core <sup>TM</sup> i5-8265U, CPU 1.80 GHz
Operating system	Windows 10 professional
Simulator	iFogSim
Simulator version	3.0.3
Memory	4 GB
Programing language	Java
Development Kit	JDK13.0.1 (64-bit)
IDE	Eclipse 4.14

### 6.3 Evaluation Results

#### 6.3.1 Loop Delay

We used this metric to measure the end-to-end loop delay. To compute the loop delay, we first determine the execution delay of each computed as follows [4]:

$$Delay_i = \beta_i - \alpha_i, \forall_i \in \mathbb{T}, \quad (1)$$

Where  $\alpha_i$  represents the starting time of execution and  $\beta_i$  represents the ending time of execution for  $i^{th}$  tasks. The average CPU time taken by all the tasks and a time taken by a particular task is computed as follows:

$$\bar{T}_{cpu} = \frac{\alpha_i \times N + \beta_i - \alpha_i}{N + 1} \quad (2)$$

Where  $N$  represents the total number of executable tasks in task set  $\mathbb{T}$ . Fig. 4 shows the computed loop-delay experienced by all applications and is measured in milliseconds. In Fig. 4, the number of nodes is represented on the horizontal axis, while the application loop delays are represented on the vertical axis.

Dynamic resource allocation and load balancing are an important methods for the optimization of system performance during application provisioning. However, the methods are widely ignored by the existing studies Fig. 4 demonstrate that proposed algorithm optimises the system performance by applying dynamic resource allocation and load balancing methods during application provisioning. This ensures the appropriate utilization of resources to reduce the overhead delay, energy and network consumption of nodes.

The main advantage of this technique is the appropriate utilization of fog computing resources which have a significant impact on system parameters. The study proved that simulation results, as shown in Fig. 4, gained optimal performance in dynamic resource allocation and load balancing method and the overall end-to-end delay of the applications are improved in the proposed algorithm framework.

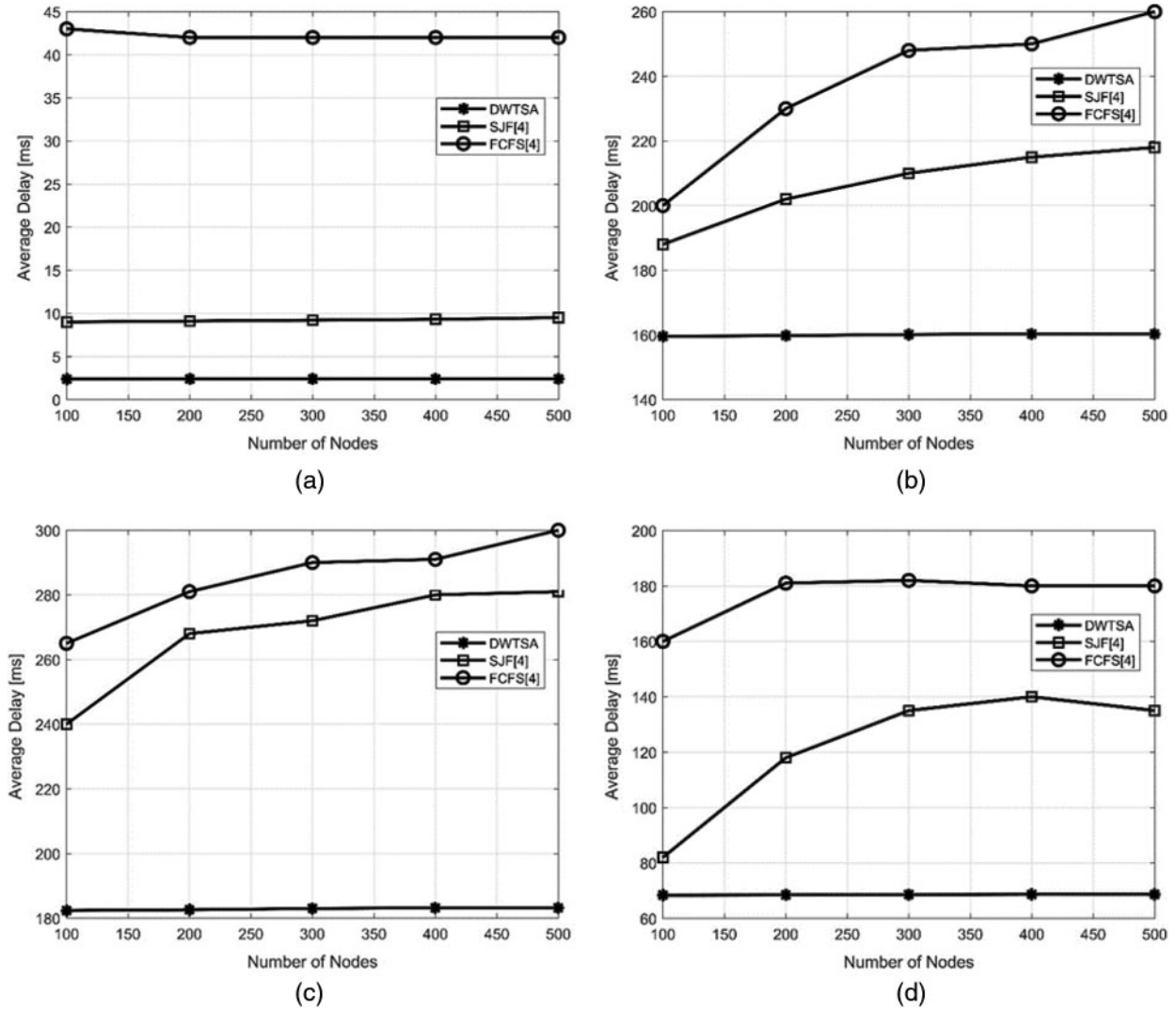
#### 6.3.2 Energy Consumption

The energy consumption of a fog device is computed as follows:

$$\mathcal{P}_c = \frac{c(\mathcal{P}_{max} - \mathcal{P}_{idle})}{c_{max}} \quad (3)$$

Where  $\mathcal{P}_{idle}$ ,  $\mathcal{P}_{max}$ , and  $c_{max}$  are represented as idle power, maximum power, and maximum capacity. The value of the term  $\mathcal{P}_{idle}$  cannot be ignored because it consumes a massive amount of energy.

The energy consumption has a great impact on system performance and usually the energy consumption of devices remains stable during static allocation. However, the dynamic and runtime environment, it creates a huge significance during task provisioning and resource allocation. The study proved that simulation results gained optimal performance of system during dynamic resource allocation and load balancing as shown in Fig. 5. The overall energy consumption of IoHT applications are improved in the proposed algorithm.



**Figure 4:** Loop delay for (a) emergency alert system (b) patients appointment system (c) patients record system (d) history of the number of nodes

### 6.3.3 Network Consumption

We evaluate the network consumption as  $\eta_{con}$ . The network consumption increase proportionally by increasing the number of devices, and thus it causes network congestion. Cloud computing has poor network performance due to heavy network congestion. Fog computing reduces network congestion by distributing the load among fog nodes. This study proposed dynamic resource allocation and load balancing algorithm for maximum utilization of fog resources. The primary goal of the proposed work is to reduce network burden on nodes. The mathematical equation of network consumption is derived as follows [4]:

$$\eta_{con} = \sum_{i=1}^N Li \times Ni \quad (4)$$

Where  $N$  represents the number of tasks,  $L_i$  represents the latency, and  $N_i$  is used for network size of  $i^{th}$  task. Fig. 4 shows the comparison result of network consumption of our proposed technique with first come first serve (FCFS) and shortest job first (SJF) algorithms.

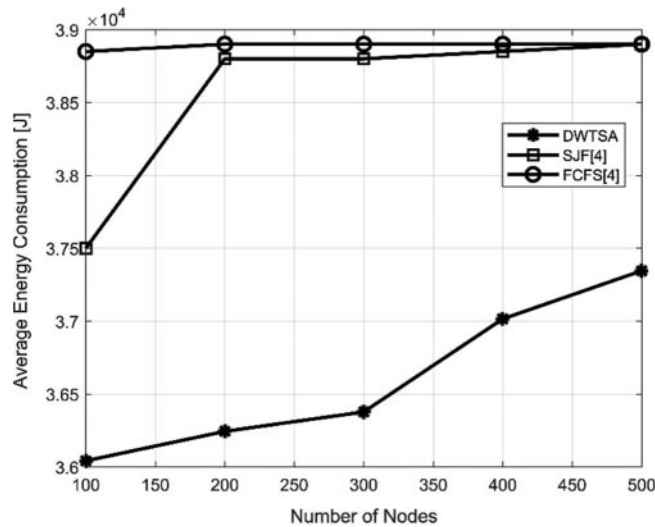


Figure 5: Average energy consumption in [J] by varying the number of fog nodes

As the number of requests increases, the network consumption also increases due to limited nodes. However, the dynamic resource allocation and load balancing algorithm makes the proper utilization of fog resources and the simulation results prove that the network consumption of the proposed algorithm are improved as depicted in Fig. 6.

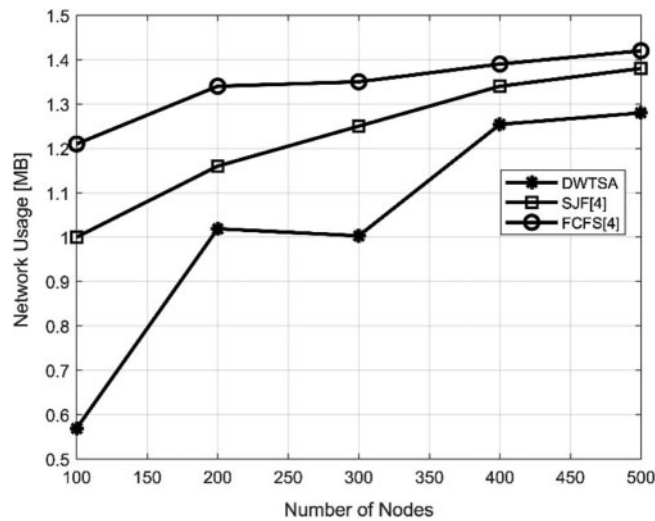


Figure 6: Network consumption in [MB]

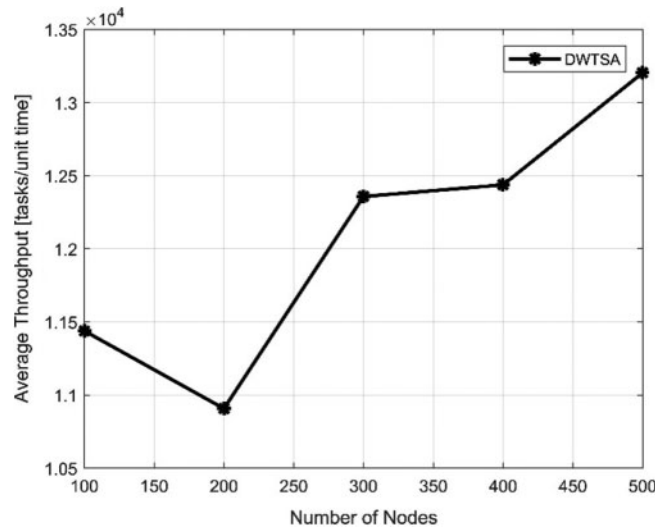
### 6.3.4 Average Throughput

Average throughput  $\mathcal{TP}_{avg}$  is measured as the total task executed by fog devices per unit time. Therefore, the average throughput utility for the number of fog devices is.

$$\mathcal{TP}_{avg} = \sum_{i \in I} \alpha i - n \gamma i \quad (5)$$

Where  $\alpha i$  is the task execution by fog server for  $i^{th}$  task,  $\gamma i$  is the  $i^{th}$  task admitted in fog queue, and  $n$  is the normalization parameter in the utility function.

The average throughput utility increases with the increase of fog devices executing the IoT tasks. The graph tilted a little as the nodes from 100 to 200, but it increased when the nodes also increased from 200 to 500. Therefore, as the fog devices increase, they are in a better position to execute as many tasks as possible as shown in Fig. 7.



**Figure 7:** Average throughput utility of Fog devices

This work has some limitations related to healthcare applications and systems, which can be improved in future work. (i) This work did not consider the security aspect for user satisfaction. (ii) This work lacks the mechanisms to find the cost model i.e., computational and application costs. (iii) This work did not consider the run time failure occurrence of any node or tasks. (iv) To find out the uncertainty of applications in a large searching space, this algorithm lacks the said problem. All of the mentioned limitations should be improved in future work.

### 6.3.5 Results and Discussion

Various IoT sensors generate different IoHT applications. Initially, the tasks are sequenced, and all the fog nodes are sorted according to the requirements of IoHT tasks. The three modules which we represented in our case study are the main working modules. The proposed algorithm is designed to make sure of the participating modules. The existing studies ignore dynamic resource allocation and load balancing mechanisms, which ultimately cause overhead delays, wastage of resources, and high energy and network consumption. The key feature of the proposed algorithm is to utilize fog resources as many as possible and reduced the overall impact of resource utilization versus IoHT applications in terms of end-to-end delay, energy, network consumption and execution time. This algorithm also

features evidence of achieving the maximum throughput, which is an additional finding of our study. After obtaining the results, the proposed algorithm performs much better compared to the algorithms used in previous work. The obtained results for healthcare applications were achieved under given conditions using iFogSim.

### 6.3.6 Performance Comparison

The performance of the proposed work compared with recent related work published in the literature is listed in [Tab. 3](#). The comparison is made based on delay, energy and network consumption, execution time, average throughput as performance metrics for research consideration.

**Table 3:** Comparison table

Reference	Approach	Domain	Metrics				
			Delay	Energy consumption	Network consumption	Execution time	Average throughput
[4]	Static	Healthcare	✓	✓	✓	✗	✗
[8]	Dynamic	Healthcare	✓	✗	✗	✓	✗
[13]	Static	Healthcare	✓	✗	✓	✗	✗
[14]	Dynamic	Vehicle	✗	✗	✗	✗	✗
[15]	Dynamic	Healthcare	✗	✓	✗	✗	✗
[16]	Dynamic	Healthcare	✗	✗	✗	✗	✗
[17]	Adaptive	NLP	✗	✗	✗	✗	✗
[18]	Adaptive	Healthcare	✗	✗	✗	✗	✗
[21]	Adaptive	Unknown	✓	✗	✗	✓	✗
[23]	Dynamic	Unknown	✓	✗	✓	✓	✗
[28]	Dynamic	Healthcare	✗	✗	✗	✗	✗
[29]	Static	Farming	✓	✓	✗	✗	✗
Proposed work	Dynamic	Healthcare	✓	✓	✓	✓	✓

## 7 Conclusion and Future Work

This paper discussed the dynamic resource allocation and load balancing of IoHT applications in fog computing environment. In this regard, we proposed a smart and appropriate technique, workload-aware dynamic resource allocation, and load balancing algorithm framework. The proposed algorithm consists of different components, such as task sequencing, dynamic resource allocation, and load balancing. Initially, the study designed the setup of an n-tier simulation environment for various IoHT applications. Subsequently, we implemented the proposed resource allocation and load balancing technique in the experimental setting and compared them with the existing methods. The results and discussion showed that the proposed approach outperforms the existing techniques in terms of delays, energy, network consumption, execution time, and average throughput utility. The proposed work considers the healthcare case study using three modules. These modules mainly focus on task sequencing, dynamic resource allocation, and load balancing for delay-sensitive applications to optimize the overall system performance, as shown in results and discussion. There are a few

limitations of the current study, such as finding the overall computational cost. In addition, the security and privacy are also the limitations of the current study.

Besides the current implication of using the proposed technique for healthcare applications, the main advantage of using the proposed method under the given healthcare scenario is that it will reduce the delay or patients' request response to save their lives and mortality. This method can be considered for other application domains like smart cities, Internet of vehicular things and smart farming purposes.

In the future, we will consider the problems mentioned earlier and use heuristic and meta-heuristic optimization techniques for dynamic workload healthcare scenarios. The current application scenario considers critical, less critical, and non-critical tasks. The future work focuses mainly on more critical tasks to optimally utilize the fog resources and minimize delay, energy, and response time.

**Funding Statement:** This research is supported and funded by King Khalid University of Saudi Arabia under the Grant Number R.G.P. 1/365/42.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

- [1] S. Tong, Y. Liu, M. Cheriet, M. Kadoch and B. Shen, "User centric user association and resource allocation in fog computing networks," *IEEE Access*, vol. 8, no. 1, pp. 671–685, 2020.
- [2] G. Li, Y. Liu, J. Wu, D. Lin and S. Zhao, "Methods of resource scheduling based on optimized fuzzy clustering in fog computing," *Sensors*, vol. 19, no. 9, pp. 1–16, 2019.
- [3] L. Yu, K. Sapra, H. Shen and L. Ye, "Cooperative end-to-end traffic redundancy elimination for reducing cloud bandwidth cost," *IEEE Transactions on Parallel & Distributed Systems*, vol. 28, no. 2, pp. 446–461, 2017.
- [4] B. Jamil, M. Shojafar, I. Ahmed, A. Ullah, K. Munir *et al.*, "A job scheduling algorithm for delay and performance optimization in fog computing," *Concurrency and Computation: Practice and Experience*, vol. 32, no. 7, pp. 1–13, 2020.
- [5] F. Bonomi, R. Milito, J. Zhu and S. Addepalli, "Fog computing and its role in the internet of things," in *Proc. Mobile Cloud Computing*, Helsinki, Finland, pp. 13–16, 2012.
- [6] A. Kumari, S. Tanwar, S. Tyagi and N. Kumar, "Fog computing for healthcare 4.0 environment: opportunities and challenges," *Computers & Electrical Engineering*, vol. 72, pp. 1–13, 2018.
- [7] Y. Deng, Z. Chen, D. Zhang and M. Zhao, "Workload scheduling toward worst-case delay and optimal utility for single-hop fog IoT architecture," *IET Communications*, vol. 12, no. 17, pp. 2164–2173, 2018.
- [8] M. Ijaz, G. Li, H. Wang, M. Sherbeeney, M. Awelisah *et al.*, "Intelligent fog-enabled smart healthcare system for wearable physiological parameter detection," *Electronics*, vol. 9, no. 12, pp. 1–32, 2020.
- [9] K. Awaisi, S. Hussain, M. Ahmed, A. Khan and G. Ahmed, "Leveraging IoT and fog computing in healthcare systems," *IEEE Internet of Things Magazine*, vol. 3, no. 2, pp. 52–56, 2020.
- [10] S. Tuli, S. Tuli, G. Wander, P. Wander, S. Gill *et al.*, "Next generation technologies for smart healthcare: challenges, vision, model, trends and future directions," *Internet Technology Letters*, vol. 3, no. 2, pp. 1–6, 2020.
- [11] M. Aazam and E. Huh, "Leveraging fog computing for emergency alert service," in *Proc. IEEE Pervasive Computing and communication Workshop*, St. Louis, MO, USA, pp. 518–523, 2015.
- [12] A. Rahmani, T. Gia, B. Negash, A. Anzanpour, I. Azimi *et al.*, "Exploiting smart e-health gateways at the edge of healthcare internet-of-things: A fog computing approach," *Future Generation Computer Systems*, vol. 78, no. 2, pp. 641–658, 2018.



- [13] S. Hassan, I. Ahmad, S. Ahmad, A. Alfaify and M. Shafiq, "Remote pain monitoring using fog computing for e-healthcare: An efficient architecture," *Sensors*, vol. 20, no. 22, pp. 1–21, 2020.
- [14] A. Lakhan, S. Memon, Q. Mastoi, M. Elhoseny, M. Mohammed *et al.*, "Cost-efficient mobility offloading and task scheduling for microservices IoVT applications in container-based fog cloud network," *Cluster Computing*, vol. 1, no. 1, pp. 1–23, 2021.
- [15] A. Lakhan, Q. Mastoi, M. Elhoseny, S. Memon and M. Mohammed, "Deep neural network-based application partitioning and scheduling for hospitals and medical enterprises using IoT assisted mobile fog cloud," *Enterprise Information Systems*, vol. 1, no. 1, pp. 1–23, 2021.
- [16] A. Lakhan, M. Mohammed, A. Rashid, S. Kadry, T. Panityakul *et al.*, "Smart-contract aware ethereum and client-fog-cloud healthcare system," *Sensors*, vol. 21, no. 12, pp. 4093–4114, 2021.
- [17] M. Mohammed, A. Ibrahim and O. Salman, "Adaptive intelligent learning approach based on visual anti-spam email model for multi-natural language," *Journal of Intelligent Systems*, vol. 30, no. 1, pp. 774–92, 2021.
- [18] K. Abdulkareem, M. Mohammed, A. Salim, M. Arif, O. Geman *et al.*, "Realizing an effective COVID-19 diagnosis system based on machine learning and IoT in smart hospital environment," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 1–8, 2021.
- [19] S. Ahmad, and D. Kim, "A multi-device multi-tasks management and orchestration architecture for the design of enterprise IoT applications," *Future Generation Computer Systems*, vol. 106, no. 1, pp. 482–500, 2020.
- [20] S. Ahmad, A. Khudoyberdiev and D. Kim, "Towards the task-level optimal orchestration mechanism in multi-device multi-task architecture for mission-critical IoT applications," *IEEE Access*, vol. 7, no. 1, pp. 140922–35, 2019.
- [21] S. Ahmad, S. Malik, I. Ullah, M. Fayaz, H. Park *et al.*, "An adaptive approach based on resource-awareness towards power-efficient real-time periodic task modeling on embedded IoT devices," *Processes*, vol. 6, no. 7, pp. 90–119, 2018.
- [22] S. Ahmad, S. Malik, I. Ullah, H. Park, K. Kim *et al.*, "Towards the design of a formal verification and evaluation tool of real-time tasks scheduling of IoT applications," *Sustainability*, vol. 11, no. 1, pp. 204–232, 2019.
- [23] Q. Shaheen, M. Shiraz, M. Hashmi, D. Mahmood, Z. Zhiyu *et al.*, "A lightweight location-aware fog framework (LAFF) for QoS in internet of things paradigm," *Mobile Information Systems*, vol. 1, no. 1, pp. 1–15, 2020.
- [24] P. Hu, S. Dhelim, H. Ning and T. Qiu, "Survey on fog computing: architecture, key technologies, applications and open issues," *Journal of Network and Computer Applications*, vol. 98, no. 1, pp. 27–42, 2017.
- [25] P. Bellavista, J. Berrocal, A. Corradi, S. Das, L. Foschini *et al.*, "A survey on fog computing for the internet of things," *Pervasive and Mobile Computing*, vol. 52, no. 1, pp. 71–99, 2019.
- [26] A. Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols, and applications," *IEEE Communication Survey and Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.
- [27] H. Gupta, A. Dastjerdi, S. Ghosh and R. 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*, vol. 47, no. 9, pp. 1275–1296, 2017.
- [28] J. Rajan, S. Rajan, R. Martis and B. Panigrahi, "Fog computing employed computer aided cancer classification system using deep neural network in internet of things based healthcare system," *Journal of Medical Systems*, vol. 44, no. 2, pp. 1–10, 2020.
- [29] A. Malik, A. Rahman, T. Qayyum and S. Ravana, "Leveraging fog computing for sustainable smart farming using distributed simulation," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 3300–3309, 2020.