IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received May 23, 2021, accepted June 4, 2021, date of publication July 1, 2021, date of current version July 13, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3094033

Fog Based Architecture and Load Balancing Methodology for Health Monitoring Systems

ANAM ASGHAR[1], ASSAD ABBAS[1], (Member, IEEE), HASAN ALI KHATTAK[2], (Senior Member, IEEE), AND SAMEE U. KHAN[3], (Senior Member, IEEE)

[1]Department of Computer Science, COMSATS University at Islamabad, Islamabad 45500, Pakistan
[2]Department of Computing, School of Electrical Engineering and Computer Science, National University of Sciences and Technology (NUST), Islamabad 44000, Pakistan
[3]Department Electrical and Computer Engineering (ECE), Mississippi State University (MSU), Starkville, MS 39762, USA

Corresponding author: Hasan Ali Khattak (hasan.alikhattak@seecs.edu.pk)

ABSTRACT With the increased number of data and data-generating devices in healthcare settings, the health monitoring systems have started to experience issues, such as efficient processing and latency. Several health-monitoring systems have been designed using Wireless Sensors Networks (WSN), cloud computing, fog computing, and the Internet of Things (IoT). Most of the health monitoring systems have been designed using the cloud computing architecture. However, due to the high latency introduced by the cloud-based architecture while processing massive volumes of data, large-scale deployment of latency-sensitive healthcare applications is restricted. Fog computing that places computing servers closer to the users addresses the latency problems and increases the on-demand scaling, resource accessibility, and security dramatically. In this paper, we propose a fog-based health monitoring system architecture to minimize latency and network usage. We also present a new Load Balancing Scheme (LBS) to balance the load among fog nodes when the health monitoring system is deployed on a large scale. To validate the effectiveness of the proposed approach, we conducted extensive simulations in the iFogSim toolkit and compared the results with the cloud-only implementation, Fog Node Placement Algorithm (FNPA), and LoAd Balancing (LAB) scheme, in terms of latency and network usage. The proposed implementation of the health monitoring system significantly reduces latency and network usage compared to cloud-only, FNPA, and LAB Scheme.

INDEX TERMS Internet of Things (IoT), fog computing, health monitoring system, load balancing.

I. INTRODUCTION

The Internet of Things (IoT) has gradually become an integral part of the human life [1]. The IoT has shown its significance and potential in several domains, such as smart cities, smart home systems, healthcare systems and so on [2]. As a matter of fact, the IoT has tremendously transformed the healthcare environment, like other application areas. The use of IoT technology in healthcare aims to improve the efficiency of medical care by automating human-led activities [3]. The key goal of healthcare applications is to constantly monitor a patient's health condition. Real-time and time-sensitive treatments, therefore, have a major role to play in healthcare. Several healthcare architectures have been proposed, among which mostly are designed by integrating the IoT with cloud computing. Cloud computing has evolved and became a feasible solution for data processing and storage applications [4]. Cloud computing, however, suffers from significant challenges, such as delay in data transmission, processing of massive amounts of data and traffic overcrowding, etc. These issues are primarily caused due to placement of cloud servers at large physical distances from the IoT devices [5].

As is evident from the critical nature of the domain, healthcare applications cannot afford the delays. Therefore, it is not feasible to use traditional cloud computing services to acquire and analyze the medical data of patients over a broad geographical area because it not only involves significant communication delays but also has high network usage [6]. To that end, fog computing has emerged as a new paradigm to solve the aforementioned underlying challenges of traditional cloud based computing [7]. Fog nodes are the computing devices placed at physically dispersed locations. Various

The associate editor coordinating the review of this manuscript and approving it for publication was Muhammad Khurram Khan.

heterogeneous devices are linked to the network, providing computing and storage resources in the geographically dispersed architecture of fog computing. Fog computing offers an architecture, a more adaptable and safer way to handle data, with low bandwidth utilization.

The concept of health care systems reflects the fact that in most countries, healthcare face challenges that continue to expand owing to the aging population [8], the increase in chronic illnesses in many regions, and the unavailability of medical practitioners [9]. According to recent research from the World Health Organization (WHO), the global population of people aged 60 and above is predicted to reach 2 billion by 2050, up from 900 million in 2015 [10]. Moreover, 60% of all fatalities worldwide and 85% of the fatalities in China account for chronic illness. Chronic illness expenditures represent around 75% of the United States' overall health care costs [11]. People with inadequate inaccessible medical resources struggle to live in adverse environments, with a growing number of patients being challenging, particularly in remote areas [9]. The significant increase in the number of challenges mentioned above has rendered it crucial to discover technical solutions to these challenges, especially in the field of healthcare. Wireless communications, through technological advancements in health monitoring systems and the IoT, will make a substantial contribution to enhance performance and lowering healthcare expenses. IoT encourages these patients to be monitored, by providing low-cost home monitoring systems to detect early signs of worsening health and to provide more quick response and treatment. Therefore, in this paper, we propose a fog computing-based architecture for the health monitoring environment.

The proposed fog-based health monitoring architecture provides information about the patients' health status by processing the data sensed and transmitted by the sensors. The proposed architecture consists of three tiers. The sensors attached to the patients to detect and transmit the vital signs, such as body temperature, heart rate, and pulse rate, etc. constitute the first tier of architecture. The intermediate layer is the fog layer containing the fog nodes, co-located with the Base Stations (BS), to which all the IoT devices are connected. The IoT devices in first tier transmit the sensor-generated data to the fog nodes, which process the data stream in order to diagnose whether the patient is in a critical state or not and pass the results to store in the cloud server via a proxy server placed in the top layer. The fog nodes also send the results of the patient's health status back to the patient's smartphones to get them displayed. The proposed architecture of the health monitoring system is intended to provide patients with real-time medical assistance without interruption, and fog computing has proven its efficiency to be implemented in time-sensitive applications. Fog computing places the resources near the proximity of end-users, thus provides a mechanism for handling huge amount of data generated from end user devices. Therefore, in the system being proposed, the use of fog computing is suitable due to the need for real-time efficient data processing.

In comparison to the cloud server, fog servers have restricted computing and storage capacities. With an increasing number of user requests, data transfer in massive systems increases the load on the fog server [12]. For large scale deployment of proposed fog-based health monitoring system, the growing number of patients' requests for a single fog node will increase the load on that particular fog node. In this situation, that fog node will be overcrowded, while the remaining fog nodes will likely remain inactive, increasing response time and incurring delay. Due to time-sensitive nature of health monitoring system, we proposed a Load Balancing Scheme (LBS), which efficiently distributes the load to other neighbouring fog nodes to minimize the latency and network usage. As from [13], we assume that IoT data flows may incur traffic latency and computing latency. In the proposed LBS approach, the IoT device selects a suitable fog node in order to reduce the latency of the proposed health monitoring system. The simulations were conducted in the iFogSim toolkit to evaluate the performance of the proposed approach by comparing against several benchmark techniques, such as cloud-only implementation, Fog Node Placement Algorithm (FNPA), and LoAd Balancing Scheme (LAB Scheme). In the cloud-only implementation, whole data generated from IoT devices is transmitted to the cloud server, without the implementation of fog layer, while fog nodes are still utilized in FNPA, and LAB Scheme to execute tasks. The experimental results proved the effectiveness of the proposed technique in terms of latency and network usage. The contributions of this paper are following:

1) A fog computing-based three-tier architecture for the health monitoring system has been proposed, where fog nodes reside in the intermediate layer. The data streams of the patient's physiological parameters, generated from sensors are transmitted through smartphones, to the fog node. Fog nodes process the incoming data streams to check if the health status of patients is critical or not. The patient's health results are transmitted back to the patient's smartphone and also forwarded to the cloud for storage.
2) The Load Balancing Scheme (LBS) has been proposed to balance the load among fog nodes in health monitoring systems deployed at a large scale.
3) The performance of LBS for health monitoring is evaluated against two metrics namely, the latency and the network usage.
4) Extensive simulations are carried out in the iFogSim toolkit to show the effectiveness of the LBS against cloud-only implementation, FNPA, and LAB Scheme.
5) Experimental results show a significant reduction in latency and network usage in comparison to cloud-only and other fog computing-based implementations.

The paper is organized as follows: Section II presents the background information and motivation for this work. The recent research works related to the architecture of health monitoring systems and load balancing in fog-based systems are provided in Section III. The proposed architecture for health monitoring systems is explained in Section IV, while Section V presents the Load Balancing Scheme (LBS). Section VI presents the experimental setup, while experimental results are demonstrated in Section VII. Discussions are provided in Section VIII and finally, Section IX presents the conclusion of our research work.

II. BACKGROUND AND MOTIVATION

In a cloud-based health monitoring system, a huge amount of sensor-generated data is supposed to be transmitted to the clouds server from user IoT devices, which requires a large number of useable network resources [14]. For the time-sensitive environments like health monitoring systems, delay is a crucial parameter, and this also increases if the system is deployed on large-scale.

Fog computing is the paradigm that leads to solving the above-mentioned problems. Fog computing brings the resources near the edge of the network thus decreasing the latency. Fog computing is the extended version of cloud services, so the cloud layer cannot be replaced entirely with fog; instead, both must co-exist to assist each other whenever necessary [1]. In addition to offering local processing and storage, fog computing is capable of managing a set of devices and sensors [15]. Therefore, fog computing appears to be more suitable for the IoT systems requiring particular characteristics.

By employing fog-based computing in the system architecture, the frequent transmission of data to the cloud server decrease, which eventually minimizes the latency of the application. Many researchers for example, [16] have claimed that latency is minimized by implementing fog computing-based architecture instead of a cloud computing-based architecture. Fog computing also minimizes the traffic over the network and enhances scalability making it perfect for IoT deployments. In real-time applications, network usage is still a critical parameter and is significantly reduced by deploying the fog-enabled architectures [1].

The transmission of real-time data in large applications increases the load on fog servers with an increasing number of IoT device users. It is, therefore, necessary to balance the load on fog nodes in order to maintain applications efficiently. Load balancing refers to the application traffic allocation across various servers to improve the applications' capability and stability. If one server becomes congested due to more client requests, some of the load can be passed to the next server. Therefore, as a result of this workload distribution, optimal utilization of energy and resources can be achieved [7].

III. RELATED WORK

A fog-computing based smart healthcare system, named HealthFog, is presented by Tuli *et al.* [17] for the diagnosis of heart disease. The proposed solution integrates software-based hardware equipment and enables rapid and reliable data transmission. In [18], a fog-based framework of health surveillance, FAAL, is introduced for patients with chronic neurological disease. Patients' movement data is gathered and transferred to the fog layer where a clustering algorithm is employed to reduce the load on the fog nodes. Another fog-based eHealth application is proposed by Vilela *et al.* in [19], to monitor the health condition of patients by collecting body physiological parameters as well as environmental factors such as light, air quality, etc. Mukherjee *et al.* [20] proposed a three-tier mobility-aware Internet of Health Things (IoHT) framework, which consists of sensor nodes, fog nodes for parametric health control, and a cloud server for processing in the case of abnormal health status. In [17]–[20], researchers validated their proposed fog computing-based architectures of health monitoring systems by comparing only against the cloud computing-based implementation. Instead, we compared our proposed fog-enabled health monitoring system with two other fog-based implementations i.e. FNPA and LAB Scheme, in addition to cloud-only implementation.

Hassen *et al.* [21] presented a fog-based health tracking application in which the fog layer analyzes the sensor-generated physiological data and transmits it to the cloud using a special JSON data model REST API. As in healthcare systems, real-time data analysis is needed, therefore, Badidi and Moumane also presented architecture in [22]. However, no simulations were performed in [21], [22] to evaluate the proposed architectures. Saidi *et al.* [23] and Debauche *et al.* in [24] proposed fog-based health surveillance systems for elderly and unattended people. To monitor blood glucose levels, a fog-enabled healthcare system is proposed by Devarajan *et al.* [25], in which the J48Graft decision tree is used to forecast a higher degree of risk for diabetes. Although the researchers in [23]–[25] conducted simulations to validate their proposed systems but did not provide sufficient results about performance metrics of latency and network usage.

For the diabetic patients suffering from cardiovascular diseases, a fog computing enabled health monitoring system is proposed by Gia *et al.* [26]. Vedaei *et al.* [27] developed a framework COVID-SAFE, to minimize the corona exposure risk. The data processing and analysis is carried out on fog nodes integrated with Machine Learning (ML) tools. Hassen *et al.* [28] presented a fog computing enabled home hospitalization system. This approach enables the patients to be treated at home in convenience, where physicians can monitor patient health's environmental status. In [26]–[28], authors did not validate their proposed approaches for latency and network usage.

An IoT-enabled healthcare system is suggested by Khattak *et al.* [29] in which different fog nodes were used to manage the requests of patients from different cities. The patient's status is monitored on the fog node, and if the status of the patient is critical, the request is transmitted on a cloud server without delay; otherwise, this request is handled by the same fog node. Although various topology configurations are simulated for performance evaluation, yet results are not compared with any cloud or fog-based healthcare system implementation. Paul *et al.* [30] presented a fog-enabled health monitoring system architecture where the fog nodes process the data by using an optimized task scheduling algorithm. In [31], another cloud and fog enabled system architecture is proposed for monitoring the patient's medical conditions. To process the acquired data, tasks are efficiently distributed by the proposed task allocation strategy. The performance of the proposed architecture and algorithm in [31], [30] has been evaluated by comparing the simulation results with cloud-only architecture, however not compared against any fog-based architecture.

Hassan *et al.* [32] proposed three-layer remote pain monitoring system architecture where the fog node detects the pain with the help of the digital signal processing techniques. Compared with cloud-based implementation, the proposed approach resulted in low latency. However, as the number of patients increases, the proposed solution would not scale well. This is because the single fog node is responsible for handling all of the hospital's data. Verma *et al.* [33] proposed an algorithm for balancing load between the three layers of the proposed architecture. A threshold to control the number of activities carried out is allocated to the fog layer. After exceeding this threshold, job requests are passed to the cloud server present in the topmost layer. The simulation results demonstrated that the proposed approach reduced the average turnaround time. A Multi-tenant Load Distribution Algorithm for Fog Environments (MtLDF) is proposed by Neto *et al.* [34]. MtLDF has proven to be more efficient in distributing the load than the Delay-Driven Load Distribution (DDLD). A dynamic resource allocation method, named DRAM, has been proposed by Xu *et al.* [35] for balancing the load in fog-based systems. Through statically assigning resources and dynamic services migration, a related resource allocation approach is provided which proved to be efficient in terms of resource utilization; however, latency is not considered for its performance evaluation. An energy-efficient strategy has also been presented by Mahmoud *et al.* [36] which assigns tasks to the fog nodes according to their computing capability and power usage. It was observed by conducting simulations that the proposed approach shows an enormously beneficial impact on application latency, network use, and energy use.

For efficient resource utilization, Tun and Paing [37] proposed a Fog Node Placement Algorithm (FNPA) that associates the IoT devices with the nearest suitable fog node with adequate resources (CPU, RAM, and Bandwidth). FNPA proved to be efficient as compared to cloud-only implementation and fog-node with minimum distance approach, by showing significant reduction in latency, cost of execution, and network usage. Fan and Ansari [38], mentioned that the data flows from IoT devices may incur two types of delays namely the network delay caused by the number of service requests, and computing delay caused by resource allocation for service requests. For the hierarchical based cloud computing, the authors proposed Workload ALLocation (WALL) scheme in [38] and Application-awaRE workload Allocation (AREA) scheme in [39]. Both of the presented schemes assign IoT users' requests to suitable cloudlets to reduce network delay and computing delay, which ultimately reduce average response time of the application. In [13], Fan and Ansari presented a LoAd Balancing Scheme (LAB Scheme) for fog computing-based systems in order to reduce the IoT dataflow latency. LAB Scheme provides a solution by connecting the user devices to the appropriate BSs for workload allocation.

Many prior works [17]–[20], [30], [31], [36] compared the simulation results of their proposed fog-based approaches with cloud-based implementation only. Other works [26]–[28] never considered the performance metrics such as latency and network usage for evaluation of proposed architectures. Some of the researchers consider one metric only; such as in [35], the impact of the proposed approach on latency is not discussed. While in [13], [33], [34], [38], [39], the network usage is not considered for performance evaluation. Some of the prior works [22], [21] never conducted experiments to validate the performance of their proposed approaches. Unlike prior work, we conducted extensive simulations with five different topology configurations and compared the results with cloud-only implementation and two other fog-based implementations i.e. FNPA and LAB Scheme. These simulations are carried out for latency and network usage parameters in which LBS outperforms cloud-only, FNPA, and LAB Scheme.

Several researchers proposed fog-based architectures for health monitoring systems that outperformed cloud-based architectures. Nevertheless, none of the researchers compared their proposed approach with a fog-based architecture for the health monitoring system previously presented. Thus, it is desirable to develop a fog-based approach more effective than previously presented fog-based solutions. As is mentioned earlier that latency and network usage are crucial parameters for health monitoring systems, most of the researchers only validated their proposed approaches against latency, however, network usage was not considered as a performance metric. Therefore, it is also needed to compare our proposed approach in terms of network usage in addition to latency.

IV. PROPOSED SYSTEM ARCHITECTURE

In this section, three-tier architecture of a fog-based health monitoring system is presented. The first tier of the proposed architecture consists of sensors attached to the patient to sense the vital signs, for example, body temperature, heart rate, and pulse rate, etc., and transmit to the fog nodes through smartphones. The fog nodes make up the second tier of the architecture. The data collected from IoT devices is analyzed by the fog nodes that are co-located with the BSs, and the results of the patient's health status are sent back to the patient's smartphone. To ensure that the patients get an immediate response in real-time environment, the fog nodes are placed at the edge of network i.e. closer to the IoT devices. The results of patient's health status are also stored in the datacenter present at cloud layer i.e. the top layer of the proposed architecture. The communication link between cloud server and fog nodes is built via a proxy server. The cloud server is primarily configured in this architecture to provide large data centers for storage. The fog computing-based architecture of our proposed health monitoring system is shown in Figure 1.

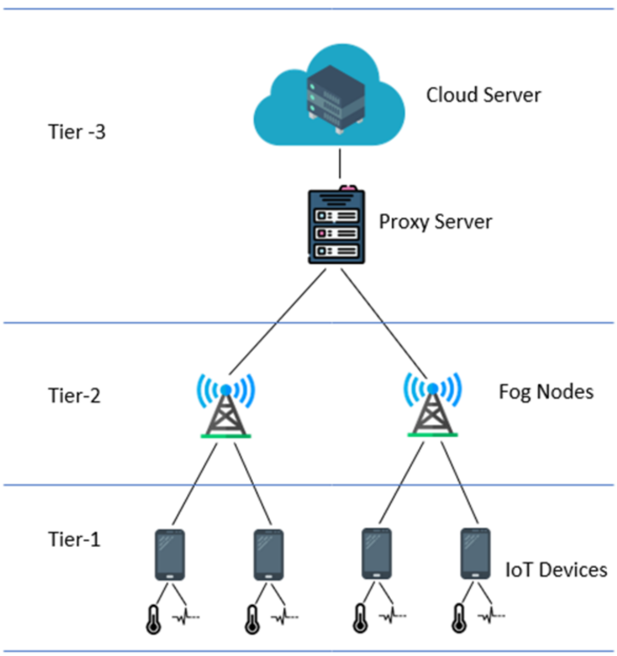

FIGURE 1. Three-Tier Architecture of Fog based health monitoring system.

A. IOT LAYER

The IoT devices form tier-1 of the proposed health monitoring system. The sensors connected with smartphones record the body temperature, heart rate, and pulse rate of the patient and send this data to an upper-level fog device to process and check whether the patient is in critical condition or not.

B. FOG LAYER

This intermediate layer is situated between the IoT and the cloud layer. The data generated from sensors is forwarded to the fog layer for processing. There are Base Stations (BSs), to which fog nodes are attached and the coverage area of BSs may overlap [13], as shown in Figure 2. In this way, each IoT device is associated with only one fog node or BS but may be placed in the coverage area of more than one BSs. The fog nodes are less capable than a cloud in terms of networking, computing power and storage space. The fog layer serves as a supporting intermediate layer, to process and analyze the real-time data near the end-users, in the proposed architecture of the fog-based health monitoring system.

C. CLOUD LAYER

The cloud server and a proxy server constitute the top layer of the proposed architecture of health management system. The cloud is primarily responsible for providing additional computing and storage resources. A proxy server connects and enables the data transmission from the fog layer to the cloud server and vice versa. Following the processing of sensor-generated data, fog nodes forward the patient-health status results to the cloud server that includes a permanent database to record the results. Moreover, the data can be retrieved from the cloud at any time.

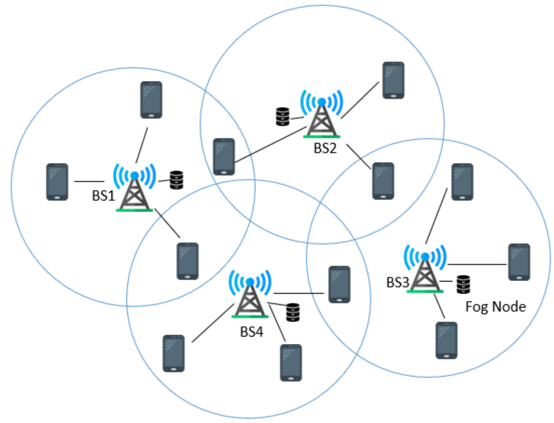

FIGURE 2. Architecture of the proposed system showing overlapping coverage areas of fog nodes/Base Stations.

D. APPLICATION MODEL

The application model of the system in fog computing is composed of a collection of application modules. Each application module is responsible for performing one specific operation on the data collected.

The application model of our proposed fog-based health monitoring system comprises of three application modules namely, (i) Client Module, (ii) Processing Module, and (iii) Storage Module. In this scenario, as the patient is supposed to be the client to the proposed health monitoring system, so the initial interface to the patient is provided by the *Client Module*. The sensor generated data of patient's health parameters, such as body temperature, heart rate, and pulse rate are collected by the appropriate sensors in the *Client Module*.

The data after being acquired by a sensor is transmitted through the patient's smartphone to the *Processing Module* in the fog node, for processing. The results of the patient's health status are sent back to the *Client Module* in smartphone to get them displayed. The *Storage Module*, responsible for record-keeping, is integrated into the cloud server. The processed results of the *Processing Module* are also transmitted to the *Storage Module*. The application model of the proposed health monitoring systems is illustrated in Figure 3.

V. LOAD BALANCING SCHEME

In our proposed architecture, Base Stations (BSs) are co-located with fog nodes, and cellular communication between devices is possible because of the BSs. The coverage areas of neighboring BSs may overlap, and end-user devices in those overlapping regions can be connected with appropriate BSs. However, any BS may become the barrier

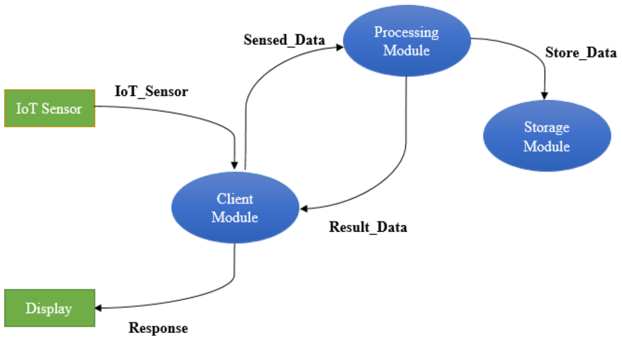

FIGURE 3. Application Model for fog-based health monitoring system.

in communication if it is overwhelmed by heavy traffic load, which reveals that the main factor of delay in overall response is the latency of dataflows. As stated earlier, that the latency of data flow on IoT devices involves two types of latencies, one is communication latency caused by traffic load, while the other is computing latency, which is due to computing load [13]. Therefore, the load balancing method must involve both the traffic and computing loads of BS and fog nodes, respectively. We have taken the situation into consideration that if BS is overloaded by connected IoT devices and causes network congestion, then some of the connected IoT devices will be offloaded to some other less loaded BSs. This will decrease the traffic load of the previous BS and may increase the computing load of the newly assigned BS. Then to reduce this computing load, some IoT devices will be offloaded to some other BS in the neighbor. In this way, the traffic load of the newly assigned BS may increase due to the balancing of the computing load of the previous BS. Here, in order to reduce the system's overall latency and network usage, we need a load balancing approach to balance both the traffic and computing loads of BSs or fog nodes. Therefore, we propose a Load Balancing Scheme (LBS) in order to minimize the latency and network usage of fog network by considering both types of latencies i.e. the communication latency and computing latency.

Some of the recent research works proposed load balancing mechanisms, one of which is the Fog Node Placement Algorithm (FNPA) [37] that associates the IoT device with the nearest fog node having maximum resources (CPU, RAM, and Bandwidth) to allocate workload. In another research work [13], LAB Scheme is presented in which appropriate fog node selection for workload allocation is made by using some complex formulas involving computing capacity of fog node and data flow rate of IoT device, etc. In our proposed approach LBS, to estimate the communication latency and computing latency of the fog network, we adopted the same procedure as researchers followed in [13]. Therefore, Equation 1, 3, 4, 5…10 have been adopted from [13]. The symbols and their interpretations used in this paper are shown in Table 1. The set of fog nodes or BSs installed in an area can be expressed as:

$$J = j_1, j_2, j_3, \dots j_n$$

TABLE 1. Key Symbols used in this paper.

Symbol	Interpretation
J	Set of Base Stations or fog nodes
D	Set of IoT devices
$P(x)$	Transmission power of IoT device at location x
$g(x)$	Channel gain of IoT device at location x
$fl(x)$	Flow arrival rate at location x
$l(x)$	Traffic size at location x
$c_j(x)$	Capacity of IoT device at location x
$v(x)$	Computing size of data flow at location x
C_j	Computing Capacity of Fog Node j
$e_j(x)$	Traffic load density of IoT device at location x
$\hat{e}_j(x)$	Computing load density of IoT device at location x
TL_j	Traffic Load of Base Station j
CL_j	Computing Load of Base Station j
L_m	Communication Latency Ratio
L_p	Computing Latency Ratio

A. TRAFFIC LOAD

Assume that x is a location in area A, where IoT device placed at x location has the transmission power as $P(x)$, channel gain as $g(x)$ and noise power denoted by σ^2. Each IoT device in the network has a Signal to Noise Ratio $SNR(x)$, which can be calculated by using Equation 1:

$$SNR(x) = \frac{P(x) \times g(x)}{\sigma^2} \tag{1}$$

where channel gain[1] $g(x)$ can be calculated in Equation 2, as follows:

$$g(x) = 10 log_{10}\left[\frac{\lambda^2}{(4\pi d)^2}\right] \tag{2}$$

It is to be noted that λ is the wavelength while the IoT device and BS are separated by the distance d. Wavelength λ can be calculated by dividing the speed of light by carrier frequency of BS. We assume that if an IoT device is linked to BS j, and BS j has the bandwidth BW_j, then the capacity of the IoT device $c_j(x)$ would be calculated as stated in Equation 3:

$$c_j(x) = BW_j \times log_{10}(1 + SNR(x)) \tag{3}$$

The IoT device at location x has some traffic load density for j^{th} BS. This traffic load density can be expressed by using Equation 4:

$$e_j(x) = \frac{fl(x) \times l(x) \times b_j(x)}{c_j(x)} \tag{4}$$

According to [13], IoT data flows follow Poisson Point Process and have an average flow rate denoted by $fl(x)$, $l(x)$ is the traffic size of the flow, $b_j(x)$ is the binary indicator if the device is associated with the respective BS or not. By adding the traffic load densities of IoT devices, we can estimate the traffic load of BS by using Equation 5:

$$TL_j = \sum_{x \in A} e_j(x) \tag{5}$$

[1] https://www.sis.pitt.edu/prashk/inf1072/Fall16/radioprop.pdf

Based on the assumptions from [13], BS j has the communication latency ratio L_m, which can be calculated in Equation 6:

$$L_m(j) = \frac{TL_j}{1 - TL_j} \tag{6}$$

B. COMPUTING LOAD

It is observed that the latency of data flows is affected by the computing latency in fog nodes. Considering $v(x)$ as the average computing size of data flow, we can calculate the computing load density of IoT device by using Equation 7:

$$\hat{e}_j(x) = \frac{fl(x) \times v(x) \times b_j(x)}{C_j} \tag{7}$$

where C_j is the computing capacity of the fog node. In order to estimate the computing load of BS j, the computing load densities of associated IoT devices would be aggregated, as shown in Equation 8.

$$CL_j = \sum_{x \in A} \hat{e}_j(x) \tag{8}$$

Based on the assumptions from [13], BS j has computing latency ratio L_p that can be calculated in Equation 9:

$$L_p(j) = \frac{CL_j}{1 - CL_j} \tag{9}$$

By using Equation 10, the Latency Ratio L of the fog network is obtained by aggregating communication latency and computing latency of all fog nodes.

$$L = \sum_{j \in J} \left[L_m(j) + L_p(j) \right] \tag{10}$$

C. FOG NODE/BASE STATION SELECTION

The traffic load and computing load of each BS are iteratively estimated, and the message is broadcast to IoT devices. The communication latency L_m and computing latency L_p of the fog, node shall be calculated following the estimation of the traffic and computing load of the BS and the fog nodes, respectively. The parameters will be checked by each IoT device, which will then choose suitable BS in each iteration with the minimum Latency value by using Equation 11.

$$m(k) = arg \min_{j \in J} \left(L_m(j) + L_p(j) \right) \tag{11}$$

Here $m(k)$ is the index of selected BS. This process will repeat until the Latency Ratio L of the overall fog network is significantly reduced. Algorithm 1 presents the mechanism for balancing the load between fog nodes.

Algorithm 1 Algorithm for Load Balancing Scheme (LBS)

```
Input: Set of IoT Devices D, Set of Fog Nodes J
Output: IoT device and Fog Node Mapping
NodeMap = [ ][ ]
repeat
    for d ∈ D do
        for j ∈ J do
            if d isInCoverageAreaOf j then
                Estimate L_m(j)
                Estimate L_p(j)
            end if
        end for
        select j with minimum (L_m + L_p)
        NodeMap.append (d,j)
    end for
    for j ∈ J do
        L = L_m (j) + L_p (j)
    end for
until L is minimized;
return NodeMap
```

VI. EXPERIMENTAL SETUP

The simulation environment used to evaluate the performance of the proposed approach is explained in this section.

The sensors, responsible for sensing body temperature, heart rate, and pulse rate, etc. frequently transmit the data to the fog nodes, via a smartphone. The data processing and analysis is carried out on the fog nodes to check the health status of the patient if he/she is in normal condition or critical condition. These fog nodes then transmit the results to the cloud for storage and also to the patient's smartphone. The connection of fog nodes with the cloud server is formed by the use a proxy server. We used the iFogSim, an open-source toolkit, to simulate and evaluate our proposed approach. It is believed that iFogSim [40] is the most effective tool for simulating fog computing-enabled applications. Several researchers [29]–[32], [36], [37] also simulated their proposed architectures in iFogSim.

Six fog nodes are created and randomly placed in a *3000 m* $\times$ *2000 m* area. Initially, each fog node has four connected IoT devices within a coverage area of a 500-meter radius. As BSs are co-located with fog nodes, placed at random places, they may have overlapping coverage areas. For simulations, IoT devices are created and attached to each BS or fog node. These IoT devices are placed at random places (using coordinate values) in the coverage area of the associated BS. To get the sensed data from sensors, the *Client Module* is embedded in the IoT devices and the *Processing Module* is created on fog nodes to check the patient's health status by processing and analyzing the incoming data. After that, the fog node sends the results to the connected IoT device, to get them displayed.

While creating the fog devices in iFogSim, we need to define values for multiple parameters, such as CPU length, RAM, Bandwidth, etc. In Table 2, the parameters used for the devices configuration in iFogSim are presented. All the computational devices created in iFogSim are known as fog devices. The computational devices, however, have different levels. The parent node on Level 0 is a cloud server. The proxy server placed at Level 1 connects the fog nodes to

TABLE 2. Values of parameters for fog-based health monitoring system.

Parameter	Cloud	Proxy	Fog	IoT Device
CPU Length (MIPS)	44800	40000	30000	20000
RAM (MB)	40000	4000	4000	4000
Uplink Bandwidth (MB)	10000	10000	10000	10000
Downlink Bandwidth (MB)	10000	10000	10000	10000

the cloud server. At Level 2, the fog nodes are placed closer to the user to provide computational and storage capacities more frequently. IoT devices are on Level 3 with sensors and actuators. Figure 4 shows the topology created in iFogSim for evaluating our proposed fog computing-based implementation. The simulations were conducted on an HP Folio 9740 EliteBook (Intel Core i5, 2.30GHz processor, 180GB SSD drive) with Windows 10 Operating System.

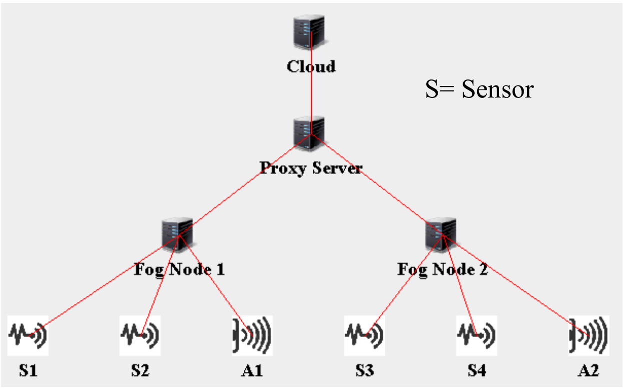

FIGURE 4. Topology of the proposed fog-based health monitoring system in iFogSim.

We simulated the proposed architecture with different topology configurations in iFogSim. We have modified the topology configurations by gradually increasing the number of IoT devices per fog node. Initially, four IoT devices were associated to each fog node, and then we increased the size of topology by increasing the number of IoT devices. The fog network topology configurations used for simulations are shown in Table 3. The performance metrics considered for evaluation of the proposed implementation are latency and network usage. By increasing the number of associated IoT devices ultimately increases the traffic as well as computing load on the fog node, leading to an increase in the network consumption and latency of that particular fog node.

TABLE 3. Scenario topologies for simulation in iFogSim.

Configurations	No. of IoT Devices	IoT Devices in Fog Network
Config-1	4	24
Config-2	6	36
Config-3	8	48
Config-4	10	60
Config-5	12	72

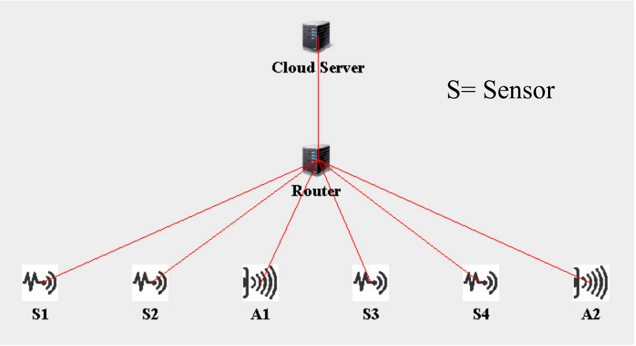

FIGURE 5. Topology of cloud-based health monitoring system in iFogSim.

Figure 5 demonstrates the topology created for cloud-only implementation in iFogSim. In this case, several devices in the IoT layer are associated via a router to the cloud server. The IoT devices send the sensor-generated data values to the cloud for analysis and the resultant information is displayed on the IoT device. The size of the topology configuration was gradually increased to inspect its impact on latency and network usage. The configuration parameters used for simulations in cloud-only implementation are presented in Table 4 while Table 5 shows the values of other parameters used in LBS.

TABLE 4. Value of parameters for cloud-only health monitoring system.

Parameter	Cloud	Router
CPU Length (MIPS)	44800	2800
RAM (MB)	40000	4000
Uplink Bandwidth (MB)	10000	10000
Downlink Bandwidth (MB)	10000	10000

TABLE 5. Value of parameters used in proposed approach LBS.

Parameters	Values
$fl(x)$	0.50 flows/second
$l(x)$	0.05 Mbits
$v(x)$	5000 CPU cycles
$P(x)$	100 mW
Cj	7.0×10^6
Uplink Frequency BW	10 MHz
Carrier frequency	2110 MHz
Noise power level	-104 dBm

VII. EXPERIMENTAL RESULTS AND DISCUSSIONS

The performance evaluation of the LBS against the cloud-only implementation, FNPA [37], and LAB Scheme [13] is carried out and results are presented in this section. The results for performance metrics of latency and network usage are provided in Table 6 and Table 7, respectively. The simulation results demonstrate that LBS minimizes the latency as compared to cloud-only implementation, FNPA, and LAB Scheme. The results also show that LBS significantly reduces

TABLE 6. Simulation results for latency.

Configurations	Latency (ms)			
	LBS	FNPA	LAB Scheme	Cloud-only
Config-1	12.81	12.57	12.86	4003
Config-2	13.22	13.41	161	4367
Config-3	121	228	358	4544
Config-4	452	508	870	4662
Config-5	1082	1148	1232	4737

TABLE 7. Simulation results for network usage.

Configurations	Network Usage (kB)			
	LBS	FNPA	LAB Scheme	Cloud-only
Config-1	341668	339724	343489	488369
Config-2	372605	378605	385639	662651
Config-3	413916	414213	432848	680546
Config-4	440857	454958	445508	699054
Config-5	466012	479232	466711	717441

network usage in comparison to the cloud-only implementation. The LBS also slightly reduces network usage as compared to FNPA and LAB Scheme.

A. ANALYSIS OF LATENCY

In time-sensitive applications like health monitoring systems, latency must be reduced. In the proposed architecture, the sensors-generated data streams are gathered and transmitted through smartphones to the fog layer for analysis. Considering a is the CPU delay for sending sensed data from sensors through smartphones, b being the time to send data towards fog node. Finally, after processing in the fog node, c is the time required to show details to the smartphone. We can calculate the latency [40] i.e. time to complete one task from source to destination, in iFogSim by using the following Equation 12.

$$Latency = a + b + c \tag{12}$$

The latency comparison of LBS with cloud-only, FNPA and LAB Scheme is presented in Figure 6. It can be observed that latency in cloud-only implementation increases significantly as the size of topology configuration grows. This is because all tasks are to be executed by the cloud, which increases load and ultimately latency. As is earlier mentioned that FNPA selects the appropriate fog node with minimum distance and maximum resources (CPU, RAM, and Bandwidth) to allocate workload. When the number of users increases and there is no more appropriate fog node with available resources, then the user requests are forwarded to the cloud server. This increases the load on the cloud server and hence latency is increased. Similarly, the LAB Scheme performs complex calculations for the selection of appropriate fog node to allocate the workload, which may take more time in performing calculations. This also increases

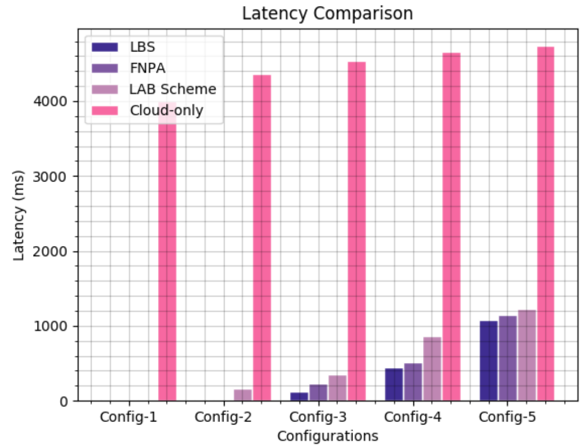

FIGURE 6. Comparison of Latency.

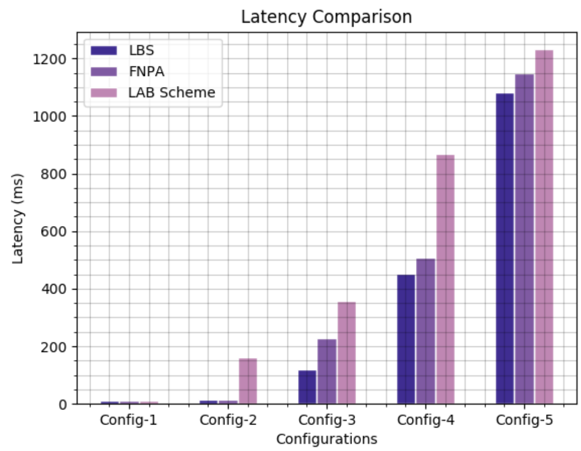

FIGURE 7. Comparison of Latency between fog-based implementations.

the latency as the size of topology configuration grows. In contrast, LBS allows the fog nodes to process all incoming tasks and also involves less complicated calculations for fog node selection. Therefore, it can be seen in Figure 6 that our proposed scheme LBS significantly reduces latency as compared to cloud-only implementation, FNPA, and LAB Scheme. To give a clear picture of the difference in results, Figure 7 shows the comparison between fog-based implementations i.e. our proposed approach LBS, FNPA, and LAB Scheme.

B. ANALYSIS OF NETWORK USAGE

By employing the cloud-only implementation of the architecture, increased volume of traffic on the cloud server leads to high network usage, due to only cloud server being responsible for all data processing. While in the case of fog-based architecture, since each fog node is supposed to process and analyze the data streams received from its connected IoT devices, the network usage decrease. Figure 8 shows the comparison of simulation results in terms of network usage in cloud-only, FNPA, LAB Scheme, and our proposed scheme LBS.

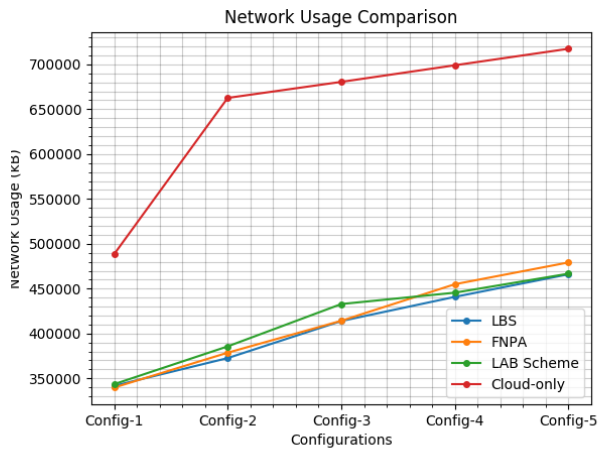

FIGURE 8. Comparison of network usage.

It can be observed that the network consumption increases in case of cloud-only implementation as the size of topology configuration grows. All of the end devices are linked via a router to the cloud server that ensures that all requests are handled by a single cloud server at a time, which increases network usage. However, there are several fog nodes in the case of fog-based implementation, in which one fog node is dedicated only for its associated IoT devices. Thus, that fog node is responsible only for processing the requests from its associated IoT devices. The LBS significantly reduces network usage in comparison to the cloud-only implementation. By conducting simulations for FNPA, it is observed that when the number of users rises and there is no more appropriate fog node with available resources, then the tasks are transferred to the cloud layer. Therefore, the network consumption increases in this case as compared to fog-based implementation but still significantly less than cloud-based implementation. While the LAB Scheme allows all processing to be done on fog nodes hence, network usage is much reduced in contrast to cloud-based implementation. As is evident that computing and analysis are primarily performed on fog servers in our proposed approach LBS, FNPA, and LAB Scheme. It can be seen in Table 7 that our proposed approach LBS reduces network usage as compared to FNPA and LAB Scheme. Although the difference is minor, still LBS proved to be efficient for deployment in a large-scale health monitoring system.

The experimental results for the performance metrics of latency and network usage demonstrate that our proposed fog computing-based implementation is an effective solution for health monitoring system. By using a fog-based architecture for a time-critical application such as health monitoring system, allows for timely retrieval of information about patient's physiological parameters, as well as a reduction in the time it takes to process data and to determine whether a patient's health condition is critical or not.

The findings help us acknowledge that fog computing has the potential to be implemented in environments where timely data processing is highly desirable. Consequently, the fog-based implementation proven to be more feasible for time-sensitive environments due to its low latency and low network usage.

VIII. DISCUSSIONS

This article presents two key findings: First, the fog computing-based architecture offers the ideal computing solution for latency-sensitive applications such as the health monitoring system. Second, our proposed approach LBS for health monitoring system is compared with the cloud-only implementation, FNPA, and LAB Scheme, in terms of two performance metrics i.e. latency and network usage, whereby the proposed approach LBS outperformed. LBS is also suitable for other fog computing-based smart home and smart city applications.

Due to the centralized nature of cloud computing, cloud servers are responsible for all processing and storage thereby increasing latency and network consumption with the growing number of user requests. Therefore, cloud computing is not an optimal solution for time-sensitive applications. Whereas fog computing introduces a new layer of fog nodes between the cloud server and end-user devices. The main feature of the fog architecture is its locally available resources that enable the system to perform essential tasks at fog nodes, thus minimizing the workload of the cloud server. The available resources at fog nodes are limited, but the distributed workload to all fog nodes results in reduced latency and less bandwidth consumption. Therefore fog-based computing might be the appropriate solution for fulfilling the QoS requirements of real-time systems such as health monitoring systems, in particular.

In this research work, our proposed fog-based approach LBS considerably decreases latency when compared to cloud-only, FNPA [37], and LAB Scheme [13]. LBS also reduces network usage than the other approaches in comparison. Although the difference in terms of network bandwidth is slight but based on overall results, our proposed approach comes out to be the preferred solution for time-critical applications. In the case of cloud-only implementation, latency and network usage increase with an increase in the number of incoming requests due to a single cloud server being responsible for the processing of all user requests. In the case of fog-based implementation i.e. FNPA, although most of the processing is performed at fog nodes, user requests are transmitted to the cloud when there are no remaining available resources at the fog layer, which ultimately increases the task completion time, thus affecting latency and network usage. Another fog-based approach, LAB Scheme performs complex and long calculations for appropriate Base Stations (BS) selection, which may increase the selection time and thus overall latency increases. In comparison to these state-of-the-art approaches, our proposed approach LBS performs proved to be efficient because of processing all tasks on appropriate fog nodes. LBS also considers both types of loads i.e. traffic load and computing load for the appropriate selection of fog nodes or BSs.

In summary, our proposed approach LBS outperformed cloud-only implementation, FNPA, and LAB Scheme in terms of latency and network bandwidth. The proposed approach LBS will enable our fog-based health monitoring system to process a large number of user requests in a timely fashion, thus providing benefits to the patients, healthcare staff, and medical practitioners. The burden at clinics will also decrease since people may monitor their health status while staying at home and must only see their doctor when health status is serious.

IX. CONCLUSION AND FUTURE WORK

Due to huge bandwidth utilization and increased latency, cloud-based computing is not an ideal environment for time-sensitive applications, such as health monitoring systems. Subsequently, fog computing has emerged as a new architecture that shifts computing resources near the edge of the network to ensure fast computing. Since fog nodes have limited computing and storage capacities, therefore load balancing among fog nodes is needed to minimize latency and ensure fast data processing. In this paper, we proposed a fog-based health monitoring system and a load balancing scheme named LBS. The proposed three-tier health monitoring system architecture provides health information to the patients by processing the sensor-generated data. Sensors and smartphones being placed in tier 1, are responsible for sensing and transmitting the patient's vital signs information to the fog nodes placed in the intermediate layer. After processing, the fog node notifies the patient about his/her medical condition and transmits the patient's health results to the cloud server for storage. For real-time systems, the number of incoming requests for each fog node may vary, which may lead to unbalance load on fog nodes. We proposed LBS to select an appropriate fog node to place incoming requests to balance the load among fog nodes. Simulations were performed to evaluate the performance of LBS with cloud-only, FNPA, and LAB Scheme. Simulation results revealed that our proposed approach outperforms cloud-only and other fog-based implementations i.e. FNPA and LAB Scheme, in terms of latency and network usage.

In the future, the proposed algorithm can be further studied and tested on larger and diverse data and can also be validated for its application for more vital signs or the diagnosis of some specific disease.

REFERENCES

[1] S. Kunal, A. Saha, and R. Amin, "An overview of cloud-fog computing: Architectures, applications with security challenges," *Secur. Privacy*, vol. 2, no. 4, p. e72, Jul. 2019.

[2] S. Kumar, P. Tiwari, and M. Zymbler, "Internet of Things is a revolutionary approach for future technology enhancement: A review," *J. Big Data*, vol. 6, no. 1, pp. 1–21, Dec. 2019.

[3] A. M. Rahmani, T. N. Gia, B. Negash, A. Anzanpour, I. Azimi, M. Jiang, and P. Liljeberg, "Exploiting smart e-Health gateways at the edge of healthcare Internet-of-Things: A fog computing approach," *Future Gener. Comput. Syst.*, vol. 78, pp. 641–658, Jan. 2018.

[4] L. Bittencourt, R. Immich, R. Sakellariou, N. Fonseca, E. Madeira, M. Curado, L. Villas, L. DaSilva, C. Lee, and O. Rana, "The Internet of Things, fog and cloud continuum: Integration and challenges," *Internet Things*, vols. 3–4, pp. 134–155, Oct. 2018.

[5] R. K. Naha, S. Garg, D. Georgakopoulos, P. P. Jayaraman, L. Gao, Y. Xiang, and R. Ranjan, "Fog computing: Survey of trends, architectures, requirements, and research directions," *IEEE Access*, vol. 6, pp. 47980–48009, 2018.

[6] B. Farahani, F. Firouzi, V. Chang, M. Badaroglu, N. Constant, and K. Mankodiya, "Towards fog-driven IoT eHealth: Promises and challenges of IoT in medicine and healthcare," *Future Gener. Comput. Syst.*, vol. 78, pp. 659–676, Jan. 2018.

[7] M. Mukherjee, L. Shu, and D. Wang, "Survey of fog computing: Fundamental, network applications, and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 1826–1857, 3rd Quart., 2018.

[8] T. M. Dall, P. D. Gallo, R. Chakrabarti, T. West, A. P. Semilla, and M. V. Storm, "An aging population and growing disease burden will require ALarge and specialized health care workforce by 2025," *Health Affairs*, vol. 32, no. 11, pp. 2013–2020, Nov. 2013.

[9] M. M. Islam, A. Rahaman, and M. R. Islam, "Development of smart healthcare monitoring system in IoT environment," *Social Netw. Comput. Sci.*, vol. 1, no. 3, pp. 1–11, May 2020.

[10] WHO. (Feb. 5, 2018). *Ageing and Health*. World Health Organization. Accessed: May, 18, 2021. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/ageing-and-health

[11] F. Meng, X. Zhang, X. Guo, K.-H. Lai, and X. Zhao, "How do patients with chronic diseases make usage decisions regarding mobile health monitoring service?" *J. Healthcare Eng.*, vol. 2019, pp. 1–7, Feb. 2019.

[12] R. A. C. da Silva and N. L. S. da Fonseca, "On the location of fog nodes in fog-cloud infrastructures," *Sensors*, vol. 19, no. 11, p. 2445, May 2019.

[13] Q. Fan and N. Ansari, "Towards workload balancing in fog computing empowered IoT," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 1, pp. 253–262, Jan. 2020.

[14] B. Farahani, M. Barzegari, F. Shams Aliee, and K. A. Shaik, "Towards collaborative intelligent IoT eHealth: From device to fog, and cloud," *Microprocessors Microsyst.*, vol. 72, Feb. 2020, Art. no. 102938.

[15] A. Mukherjee, D. De, and S. K. Ghosh, "FogIoHT: A weighted majority game theory based energy-efficient delay-sensitive fog network for Internet of health things," *Internet Things*, vol. 11, Sep. 2020, Art. no. 100181.

[16] M. Taneja and A. Davy, "Resource aware placement of IoT application modules in fog-cloud computing paradigm," in *Proc. IFIP/IEEE Symp. Integr. Netw. Service Manage. (IM)*, May 2017, pp. 1222–1228.

[17] S. Tuli, N. Basumatary, S. S. Gill, M. Kahani, R. C. Arya, G. S. Wander, and R. Buyya, "HealthFog: An ensemble deep learning based smart healthcare system for automatic diagnosis of heart diseases in integrated IoT and fog computing environments," *Future Gener. Comput. Syst.*, vol. 104, pp. 187–200, Mar. 2020.

[18] J. Vora, S. Tanwar, S. Tyagi, N. Kumar, and J. J. P. C. Rodrigues, "FAAL: Fog computing-based patient monitoring system for ambient assisted living," in *Proc. IEEE 19th Int. Conf. e-Health Netw., Appl. Services (Healthcom)*, Oct. 2017, pp. 1–6.

[19] P. H. Vilela, J. J. P. C. Rodrigues, P. Solic, K. Saleem, and V. Furtado, "Performance evaluation of a fog-assisted Iot solution for e-Health applications," *Future Gener. Comput. Syst.*, vol. 97, pp. 379–386, Aug. 2019.

[20] A. Mukherjee, S. Ghosh, A. Behere, S. K. Ghosh, and R. Buyya, "Internet of health things (IoHT) for personalized health care using integrated edge-fog-cloud network," *J. Ambient Intell. Humanized Comput.*, vol. 12, pp. 1–17, Jan. 2020.

[21] H. Ben Hassen, W. Dghais, and B. Hamdi, "An E-health system for monitoring elderly health based on Internet of Things and fog computing," *Health Inf. Sci. Syst.*, vol. 7, no. 1, pp. 1–9, Dec. 2019.

[22] E. Badidi and K. Moumane, "Enhancing the processing of healthcare data streams using fog computing," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jun. 2019, pp. 1113–1118.

[23] H. Saidi, N. Labraoui, A. A. A. Ari, and D. Bouida, "Remote health monitoring system of elderly based on Fog to Cloud (F2C) computing," in *Proc. Int. Conf. Intell. Syst. Comput. Vis. (ISCV)*, Jun. 2020, pp. 1–7.

[24] O. Debauche, S. Mahmoudi, P. Manneback, and A. Assila, "Fog IoT for health: A new architecture for patients and elderly monitoring," *Procedia Comput. Sci.*, vol. 160, pp. 289–297, Jan. 2019.

[25] M. Devarajan, V. Subramaniyaswamy, V. Vijayakumar, and L. Ravi, "Fog-assisted personalized healthcare-support system for remote patients with diabetes," *J. Ambient Intell. Humanized Comput.*, vol. 10, no. 10, pp. 3747–3760, Oct. 2019.

[26] T. Nguyen Gia, I. B. Dhaou, M. Ali, A. M. Rahmani, T. Westerlund, P. Liljeberg, and H. Tenhunen, "Energy efficient fog-assisted IoT system for monitoring diabetic patients with cardiovascular disease," *Future Gener. Comput. Syst.*, vol. 93, pp. 198–211, Apr. 2019.

[27] S. S. Vedaei, A. Fotovvat, M. R. Mohebbian, G. M. E. Rahman, K. A. Wahid, P. Babyn, H. R. Marateb, M. Mansourian, and R. Sami, "COVID-SAFE: An IoT-based system for automated health monitoring and surveillance in post-pandemic life," *IEEE Access*, vol. 8, pp. 188538–188551, 2020.

[28] H. Ben Hassen, N. Ayari, and B. Hamdi, "A home hospitalization system based on the Internet of Things, fog computing and cloud computing," *Informat. Med. Unlocked*, vol. 20, 2020, Art. no. 100368.

[29] H. A. Khattak, H. Arshad, S. U. Islam, G. Ahmed, S. Jabbar, A. M. Sharif, and S. Khalid, "Utilization and load balancing in fog servers for health applications," *EURASIP J. Wireless Commun. Netw.*, vol. 2019, no. 1, pp. 1–12, Dec. 2019.

[30] A. Paul, H. Pinjari, W.-H. Hong, H. C. Seo, and S. Rho, "Fog computing-based Iot for health monitoring system," *J. Sensors*, vol. 2018, pp. 1–7, Oct. 2018.

[31] R. M. Abdelmoneem, A. Benslimane, E. Shaaban, S. Abdelhamid, and S. Ghoneim, "A cloud-fog based architecture for IoT applications dedicated to healthcare," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–6.

[32] S. R. 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, p. 6574, Nov. 2020.

[33] M. Verma, N. Bhardwaj, and A. K. Yadav, "Real time efficient scheduling algorithm for load balancing in fog computing environment," *Int. J. Inf. Technol. Comput. Sci.*, vol. 8, no. 4, pp. 1–10, Apr. 2016.

[34] E. C. P. Neto, G. Callou, and F. Aires, "An algorithm to optimise the load distribution of fog environments," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2017, pp. 1292–1297.

[35] X. Xu, S. Fu, Q. Cai, W. Tian, W. Liu, W. Dou, X. Sun, and A. X. Liu, "Dynamic resource allocation for load balancing in fog environment," *Wireless Commun. Mobile Comput.*, vol. 2018, pp. 1–15, Apr. 2018.

[36] M. M. E. Mahmoud, J. J. P. C. Rodrigues, K. Saleem, J. Al-Muhtadi, N. Kumar, and V. Korotaev, "Towards energy-aware fog-enabled cloud of things for healthcare," *Comput. Electr. Eng.*, vol. 67, pp. 58–69, Apr. 2018.

[37] K. N. Tun and A. M. M. Paing, "Resource aware placement of IoT devices in fog computing," in *Proc. Int. Conf. Adv. Inf. Technol. (ICAIT)*, Nov. 2020, pp. 176–181.

[38] Q. Fan and N. Ansari, "Workload allocation in hierarchical cloudlet networks," *IEEE Commun. Lett.*, vol. 22, no. 4, pp. 820–823, Apr. 2018.

[39] Q. Fan and N. Ansari, "Application aware workload allocation for edge computing-based IoT," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 2146–2153, Jun. 2018.

[40] H. Gupta, A. V. Dastjerdi, S. K. 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," *Softw., Pract. Exper.*, vol. 47, no. 9, pp. 1275–1296, Sep. 2017.

ANAM ASGHAR received the bachelor's degree in software engineering from Fatima Jinnah Women University, Rawalpindi, Pakistan, in 2017. She is currently pursuing the master's degree in software engineering with the Department of Computer Science, COMSATs University at Islamabad, Islamabad, Pakistan.

ASSAD ABBAS (Member, IEEE) received the Ph.D. degree from North Dakota State University (NDSU), USA. He has been with COMSATS University at Islamabad (CUI), Islamabad, Pakistan, since 2004. He is currently working as an Assistant Professor with the Department of Computer Science, COMSATS University at Islamabad (CUI). His main research interests include but not limited to smart health, big data analytics, recommendation systems, cloud and fog computing, and social network analysis. His research has appeared in several prestigious journals, such as IEEE Transactions on Cybernetics, IEEE Transactions on Cloud Computing, IEEE Transactions on Dependable and Secure Computing, IEEE Transactions on Sustainable Computing, IEEE Systems Journal, IEEE Journal of Biomedical and Health Informatics, *ACM Computing Surveys*, *IEEE IT Professional*, *Journal of Parallel and Distributed Computing*, *Pervasive and Mobile Computing*, and *Future Generation Computer Systems*.

HASAN ALI KHATTAK (Senior Member, IEEE) received the B.C.S. degree in computer science from the University of Peshawar, Peshawar, Pakistan, in 2006, the master's degree in information engineering from the Politecnico di Torino, Torino, Italy, in 2011, and the Ph.D. degree in electrical and computer engineering from the Politecnico di Bari, Bari, Italy, in April 2015. He has been serving as an Associate Professor for the School of Electrical Engineering and Computer Science, National University of Sciences and Technology, Pakistan, since October 2020. He is currently involved in several funded research projects in various domains, such as healthcare information management, semantic Web of Things, and fog computing, while exploring ontologies and other semantic Web technologies. He has worked on Contiki OS, NS 2/3, and Omnet++ frameworks. His perspective research areas are the application of machine learning and data sciences for improving and enhancing quality of life in smart urban spaces, especially in the healthcare and transportation domain through knowledge management, predictive analysis, and visualization. His current research interests include future Internet architectures, such as the Web of Things and leveraging data sciences and social engineering for future smart cities. He is a Professional Member of ACM, and a member of societies, such as the IEEE ComSoc, the IEEE VTS, and the Internet Society. Along with publishing in good research venues and completing successful funded National and International funded projects, he is also serving as a Reviewer for reputed venues, such as IEEE Access, *IEEE Network Magazine*, *IEEE Consumer Electronics Magazine*, Hindawi, SAI, IET, and a few national publishers.

SAMEE U. KHAN (Senior Member, IEEE) received the Ph.D. degree from The University of Texas, in 2007. He was a Cluster Lead of computer systems research with the National Science Foundation and a Walter B. Booth Professor with North Dakota State University, from 2016 to 2020. He is currently the Head of the Department and the James W. Bagley Chair of electrical and computer engineering with Mississippi State University (MSU). His work has appeared in over 400 publications. His research interests include optimization, robustness, and security of computer systems. He is also an Associate Editor-in-Chief of the *IEEE IT Professional*, and an Associate Editor of IEEE Transactions on Cloud Computing, *Journal of Parallel and Distributed Computing*, and *ACM Computing Surveys*.