Multi-Objective Load-balancing Strategy for Fog-driven Patient-Centric Smart Healthcare System in a Smart City

Gaurav Goel
Department of Computer Applications, Government Engineering College, Ajmer, BTU, Bikaner, India
gaurav.itindia@gmail.com (corresponding author)

Amit Kr Chaturvedi
Department of Computer Applications, Government Engineering College, Ajmer, BTU, Bikaner, India
amit0581@gmail.com

Received: 5 May 2024 | Revised: 23 May 2024, 8 June 2024, and 17 June 2024 | Accepted: 18 June 2024

 DOI: https://doi.org/10.48084/etasr.7749

ABSTRACT

The spatially concentrated architecture of the cloud environment causes excessive latency and network congestion in traditional smart healthcare systems designed for smart cities. Fog computing underpins IoT-enabled smart city solutions for latency sensitivity by putting computing power closer to the network boundary. However, resource management issues degrade service quality and accelerate energy depletion in real-time smart healthcare systems, as the fog node workload has increased exponentially. This paper offers a fog-driven patient-centric smart healthcare system for an e-healthcare environment to maintain Quality of Service (QoS) during severe traffic load on a fog platform. The multi-objective EQLS (Energy-efficient QoS-aware Load balancing Strategy), is proposed to stabilize workload among processing nodes to increase real-time sensitivity of critical tasks within optimal response time and energy usage. Using the iFogSim simulator to present the significance of research work, the proposed technique is compared to existing load-balancing policies (Round Robin (RR) and Fog Node Placement Algorithm (FNPA)) regarding energy usage, response time, and cost. The simulation results reveal that EQLS saves 8.7% and 14.9% more energy and 6.2% and 13.4% greater response time over FNPA and RR, respectively. The results signify that the proposed approach can efficiently support real-time applications of smart cities.

Keywords-Smart Cities; Load Balancing; Wearable Sensors; Fog Computing; iFogSim

I. INTRODUCTION

Cisco predicts that by 2025, nearly two-thirds of the global population will have been online, and the Internet of Medical Things (IoMT) market will have been worthing more than $350 billion. The smart healthcare load has grown exponentially due to increased chronic diseases after COVID-19, which forced the Internet of Things (IoT) to offer a whole new range of possibilities in telemedicine services which doctors provide worldwide in smart cities relying more on social network services to predict outbreaks [1]. Such concerns stimulate developing a patient-centric system that incorporates cutting-edge technologies. By using tailored, data-driven strategies, the former seeks to improve patient outcomes, raise the standard of care, and expedite healthcare procedures.

The IoT is becoming widely accepted and increasingly used in many facets of our everyday lives. All the smart city applications, like smart parking and healthcare system components, such as people, appliances, and medication, can be continually monitored and controlled because of the pervasive computing aspect of the IoT. By automating processes that people previously completed, IoT technology may be used in healthcare to optimize both the cost and Quality of Service (QoS) of medical treatment [2]. Designing isolated wearables is no longer adequate. A whole ecosystem must be created in which body-worn web sensors will synchronize sensed data to data center services via IoT infrastructure. IoT devices may monitor daily behaviors, including blood pressure, blood glucose, and (electrocardiogram) ECG. Many severe illnesses can be avoided using IoT devices to monitor activities effectively. Medical sensors and healthcare equipment monitor patients in real-time, quickly generating data. This data are then processed, saved, and analyzed [3]. Sensor-equipped gadgets often use little power, have smaller batteries, less storage, and have less networking capacity. As a result, another framework capable of computation, storage, and analysis is required for data collection.

The transfer of services to the patient's home, which dramatically reduces the total cost of healthcare, is the main advantage of cloud computing for the health sector. Cloud services can help in early identification and effective management of health conditions. In some instances, latency-sensitive applications do not work well with cloud computing. It gets more challenging for the cloud to handle all the needs with the least latency as the number of requests and calculation demands rises. The scalability, security, data analysis, and dependability of healthcare systems offered by the IoT are greatly enhanced with fog services because cloud data centers are concentrated geographically, and sensor data processing requires multi-hop transmission, hampering the solutions' latency sensitivity [4, 5]. Fog computing provides flexible, lightweight supplemental resources closer to the end devices in healthcare systems. To finish some processes closer to the source, which would decrease latency, increase service flexibility, spread out resource requirements, and eliminate the need for multi-hop data transfer, fog supplies the usual edge computing routers, switches, and constrained computing devices, using services and management models. The working environment of a traditional Cloud Based Smart Healthcare System is shown in Figure 1.

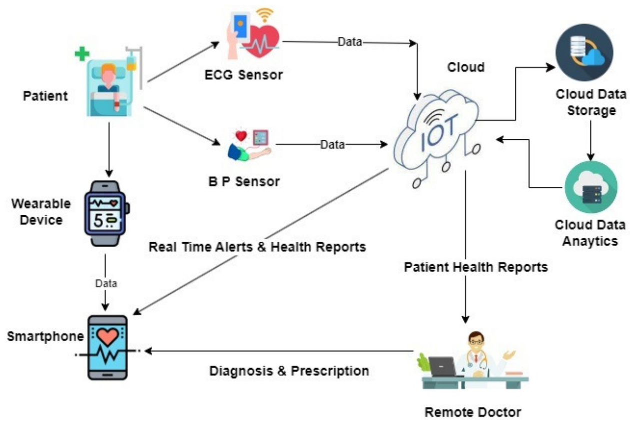

Fig. 1. Traditional cloud-based smart healthcare system.

Traditional smart healthcare systems are facing the issues of network congestion and high latency due to the geographically centralized architecture of the cloud environment. Moreover, fog computing also has specific difficulties and obstacles [6]. Devices located at the periphery of the network execute fog computing. Due to resource constraints, they are unable to process all application requests. Furthermore, heavy traffic may overwhelm the system, potentially posing a risk to patients and medics in a smart healthcare setting. This study suggests a multi-objective Energy-efficient QoS-aware Load-balancing Strategy (EQLS) to address the above challenges. This strategy aims to improve the QoS for real-time smart e-healthcare applications by distributing the workload evenly among fog nodes. It is achieved by optimizing the response time and energy consumption. This algorithm is based on the Fog Node Placement algorithm [6], which only considers the fog devices' distance and resources. The proposed approach considers the selection of the optimal cluster, the selection of the nearest resourceful fog devices, and the consideration of job priority. In addition, the energy usage performance metric is considered along with response time and cost. The main contributions of the current paper are:

- The paper introduces a fog-driven patient-centric smart healthcare system to improve the traditional cloud based smart e-healthcare systems for smart cities.
- The EQLS is proposed to enhance the service quality of real-time smart e-healthcare applications to stabilize the critical task load among fog nodes.
- Using iFogSim, a performance evaluation simulation concerning energy usage, response time, and cost is conducted for a proposed strategy to assess the patient's cardiac conditions remotely and effectively.

II. LITERATURE REVIEW

Medical sensors have lesser processing power, memory, transmission speed, and energy supply than other sensors, making IoT-based system design difficult. Streaming transmission is difficult even with short system outages. In most cases, the cloud-fog restricted resource availability and load balancing concerns make it inefficient for real-time sensitive applications regarding delay, energy, and cost. Authors in [7] proposed an approach to reduce the load on the network's nodes, improving efficiency, user experience, and productivity. It was concluded that Glowworm Swarm Optimization (GSO) is the most appropriate technique for reducing the load balancing issue after comparing its performance with those of Particle Swarm Optimization (PSO), Cuckoo Search (CKS), and GSO. Authors in [8] created a decentralized microservices guide for placing IoT applications in various resource-constrained fog settings. Microservice-related service discovery and load-balancing difficulties were also addressed. Authors in [9] reported that a weighted cost model can lessen processing time and energy utilization for IoT applications in a computing infrastructure with cloud servers, fog/edge servers, and IoT devices. By capitalizing on gateways' advantages at edge-of-the-network locations, authors in [10] provided services including real-time local storage and data processing. They showcased a smart e-health gateway prototype and advocated extending the notion of a fog network to healthcare systems by establishing a distributed smart interface layer between the sensor nodes and the cloud. Applications for the IoT were conceptualized by in [11]. The authors also tested a functional prototype that uses the patient's smartphone to share ECG securely traces collected from a customized device with other authorized parties as a fog gateway.

FOCAN, a smart city network architecture that utilizes a fog network, was proposed in [12]. The former decreases latency and increases service efficiency and energy supply when used across entities of varying capacities. Beyond the conventional cloud-based framework, authors in [13] investigated the union of fog-cloud services to provide healthcare solutions based on the IoT. The approach was analyzed implementing the iFogSim simulator, considering power consumption, distributed computing, latency, and data transmission optimization. After reviewing the leading models of fog-enabled Cloud of Things (CoT) systems, authors in [14] suggested a method for deploying application modules on fog gateways that consider energy. Healthcare systems were the primary emphasis in [15]. Fog nodes, a cloud data center, and objects or sensors comprise a given architecture's three levels. A healthcare system's fog nodes could work together to provide optimal resource and job allocation, resulting in minimal latency and a high quality of service. Authors in [16] investigated the functions of cloud, fog computing, and IoT to provide end users with continuous, context-aware services. The authors developed a three-tiered patient-focused healthcare solution architecture to collect, process, and transmit real-time data. To provide healthcare as a cloud service, authors in [17] presented a fog-assisted information paradigm that uses IoT devices. In addition, the suggested design is excellent at handling patients' demands for cardiac data. A new framework called HealthFog was created in [18] to include ensemble deep learning into fog devices. It was used for an automated investigation of real-world heart diseases.

The intelligent real-time applications' high processing and communication needs can be met by dividing the load across fog nodes according to the fog-based architecture outlined in [19]. Authors in [20] proposed the energy-efficient resource allocation method for IoT-Fog computing networks to perform job offloading based on the least cost, fault identification, and rectification techniques. The introduced approach allocated a fog node and resource block for each device and linked it to one or more devices. To improve user experience and service quality, authors in [21] proposed to categorize a request-based efficient load-balancing approach. The presented method classified requests in cluster fog nodes deploying a decision tree and an efficient K-means clustering approach. Authors in [22] proposed a fog-based deadlock management method to optimize task scheduling and deadlock prediction. They presented five modules, i.e. collector, matcher, deadlock identifier, allocator, and prioritizer, to assign the best fog nodes to task requests. Authors in [23] introduced an IoT-fog-cloud application architecture and an integrated computation model for energy efficiency. A fog-enabled smart city scenario was used to perform offloading to reduce service latency and response time. Although IoT-based apps have employed novel load-balancing methodologies more in recent years, there have been relatively moderate advances in load-balancing problems for critical tasks, especially for healthcare systems. Existing contributions utilize a gateway to connect sensors and cloud with minimum gateway role. However, the introduced patient-centric smart healthcare system meets domain-specific needs by computing most requests at fog gateways and offering solutions close to patients to ensure QoS. The summary of the literature review, is presented in Table I.

When examining the recurrent problems that researchers are trying to solve with various strategies, it becomes clear that smart healthcare monitoring systems need much help with better load-balancing techniques to manage the exponential heterogeneous IoT load on the fog layer, to make the best use of their resources and energy, and continue to deliver high-quality services in the face of future demands. The critical need for multi-objective optimization has been addressed by the proposed fog-driven patient-centric smart healthcare system using multi-objective EQLS, which manages the trade-off between response time and energy consumption for best outcomes considering job priorities and selecting optimal cluster fog resources for execution of task requests.

TABLE I. LITERATURE REVIEW SUMMARY

Ref.	Highlights	Limitations	Metrics
[10]	Set up an IOT-based health monitoring solution using Early Warning Scores (EWS).	Missed to create an effective SDN paradigm that balances the resources of fog gateways.	Energy, Network Usage
[11]	Developed fog-enabled health monitoring system prototype.	Ignored the negative impact of load movement in multi-level fog architecture.	Throughput, Response Time
[12]	Designed architecture for smart city with fog support for IoT applications.	Overlooked ways to reduce fog node energy-cost tradeoff.	Energy Usage
[13]	Outlined the cloud-fog orchestration approach to balance load and service distribution.	Ignored the effect of service request interference in a fog network under high traffic loads.	Delay, Energy, Cost
[14]	A case study of remote patient monitoring is utilized.	Paid no attention to utilize cluster-based orchestration, overload prediction, and mobility patterns to find the best solutions.	Energy, Network Usage
[15]	A framework for fog-to-fog cooperation to provide QoS for healthcare applications	Ignored considering cloudlet priority while executing tasks on fog nodes.	Latency, Energy
[16]	Discussed about the potential and problems in fog for the context of healthcare 4.0.	Overlooked multi-objective optimization methods that include different QoS.	QoS
[17]	Created a IoT-Fog based smart healthcare cloud solution for cardiac patients.	Missed to enhance the QoS for low-priority tasks.	Latency, Network Usage, Energy

III. METHODOLOGY

A. The Proposed Fog-Driven Smart Healthcare System

The widespread adoption of IoT is anticipated to result in billions of more resource-restricted devices being linked to the Internet. Most of gadgets, like medical sensors that are implanted or worn, cannot save the data they produce due to resource constraints. Moving this data to a cloud to be processed is a simple design trick. The latency of the cloud connection may be substantial due to the numerous connected devices. Furthermore, these devices may not be directly compatible with the cloud architecture owing to their limited power and bandwidth. A crucial paradigm change toward a 3-tier architecture with a more responsive design is termed fog computing. Fog is an intermediary processing layer between the data center and the end devices. It enhances the credits of the cloud by offering extra services to meet the constantly changing needs of the IoT. The elements of the proposed system for the e-healthcare environment of smart cities are spread throughout the three levels, as seen in Figure 2. The proposed system uses body-worn sensors to capture health-related data, enabling the patient to monitor many indicators independently. Contextual data, such as location, date, and time, can be added to this health data. Being context-aware the introduced system makes it possible to spot odd trends and draw more accurate conclusions about the circumstances. The vital parts of the proposed system are:

- **Body-Worn Wearable Devices:** Patient-equipped wearable gadgets comprise the first logical group of devices and services. These wearable gadgets provide ubiquitous identification, sensing, and communication capabilities that make recording biological and situational signals possible. Certain wearables can analyze the traces gathered from the sensors and, if desired, may be utilized to give feedback. The wearable device's internal storage houses the raw and diagnostic data locally. These data, which are patient-owned, make up the short-term historical health information. The patient's express consent is required before some services can access the encrypted data.

- **Edge Application Nodes:** The mobile application is loaded on the patient's Android handset and connected wirelessly to the wearable gadget to obtain the data. After installing the mobile apps, the user must utilize the Bluetooth low-energy bonding to couple the mobile device and wearable devices. After finishing the pairing procedure, the mobile interface prevents the wearable device from being linked with another device. This application node is utilized for small-scale computation, like filtering the sensor's raw data and displaying results to the user after processing tasks by fog computing gateways.

- **Fog Computing Gateways:** The fog cluster, which makes up this layer, is constructed from many e-health gateways that are dispersed geographically. Distributive fog nodes are organized into hierarchical fog layers. Processing cores, memory, storage, and network bandwidth may be added to fog nodes. The IoT devices are located near lower-level application nodes, such as smartphones and set-top boxes, which often provide interfaces for wearable apps. The detected patient data can be pre-processed by a lower-level application node and sent to next-level fog nodes known as computational gateways for computation. Only some nodes in the fog environment are constantly kept computationally active. When the data load decreases, the fog nodes' computational unit may be turned off and switched on based on demand. Thus, it is possible to make the fog environment energy-efficient and scalable.

- **Back-End Cloud Data Centers:** A cloud computing platform that uses data analytics, data warehousing, and broadcasting is utilized to make up the back-end system. Smart care deploys cloud data center resources when the fog nodes are overwhelmed and services are time-sensitive. This increases its robustness and speed at handling huge loads, making data processing location-independent.

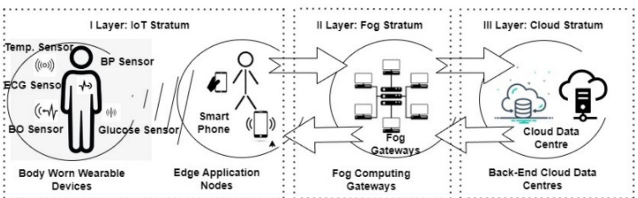

Fig. 2. The proposed fog-driven patient-centric smart healthcare system.

B. The Proposed Load Balancing Strategy

A fog node may receive health data from wearable sensors via smartphone in nearly all clusters. As the cluster node receives data, it alerts the master server node and uses the affiliated healthcare solution to determine its relevance. Based on the analysis, the cluster master server node either schedules the data for processing by computing instances of the same cluster or transmits them to the associated cluster. In addition, the cluster master server node regulates and secures communication and resource provisioning for the other nodes, distributes the load among them, keeps an eye on computing instance activity, and maintains associated meta-data. To prevent performance deterioration, the cluster master server node can also assign its duties to the other nodes in the cluster. The EQLS is proposed to allocate the resources and balance the real-time load across optimal cluster and its resourceful computing nodes to improve the quality of service of intelligent real-time applications of smart cities.

In the proposed multi-objective Algorithm 1, employing the Optimal Cluster Prediction function with the highest victory score, as evidenced in Algorithm 1.1, the task is assigned to the optimal cluster and closest available fog device to minimize the response time. Using the Identify Resourceful Gateway function with the predetermined threshold, as observed in Algorithm 1.2, a pool of fog nodes is selected for a cluster to optimize energy utilization. In the proposed strategy, the QoS is improved implementing the Shuffle Prioritise Cloudlets function displayed in Algorithm 1.3. The incoming tasks from the end devices are organized in a priority queue based on the patient critical condition defined by the network administrator. Every job that the end user submits to the balancer is added to a task queue. The job stays in the queue until the algorithm pulls it out to ensure efficient work allocation.

```
Algorithm 1: Energy Efficient QoS Aware
Load Balancing Strategy (EQLS)
Input: Array of Cluster C, List of Fog
Gateways N, Number of Cloudlet T
Output: Mapping of Cloudlet (t) & Fog Node
(n)
Energy_Efficient_QoS_Aware_Load_Balancing_
Strategy (Cloudlet[], Gateways[])
Select cluster OC = Call
(Optimal_Cluster_Prediction (C, VMs,
Thrld))
Initialize RouteMap[Edevice][Fnode] in
Sorted Order, Thrld
FogRes[] = Call
(Identify_Resourceful_Gateways (N, Thrld))
Initialize WorkLoad[], Memory[],
Network[], EnergyUsage[]
For e Ɛ RouteMap                //e=0 to
Edevice.Limit
  MapList = {}
  Etask[] = call
(Shuffle_Prioritise_Cloudlets (T))
  For Task t Ɛ Etask do         //t=0 to
Etask.Count
```

```
    For Node n Ɛ FogRes do           //n=0
to FogRes.Count
      If (t.cpu <= n.cpu && t.memory <=
n.memory && t.network <= n.network &&
t.energy <= n.energy)
      Then
        Assign t to device n
        Add t to Maplist
        Break
      Else
        Continue
      End if
    End for
    n.WorkLoad  = n.Workload + t.cpu
    n.Memory = n.Memory + t.memory
    n.Network = n.Network + t.network
    n.EnergyUsage = n.EnergyUsage +
t.energy
  End for
End for
```

Algorithm 1.1: Optimal Cluster Prediction
Input: Array of Cluster C, No of Virtual Machines VMs, Cluster Threshold Load Thrld
Output: Select Optimal Cluster OC

```
Optimal_Cluster_Prediction (C, VMs, Thrld)
Calculate each Cluster_Load []
For each cluster C_i, do
//i=0 to cluster.count
  For each VM_j Ɛ C_i
// j=0 to virtual machine.count
    Calculate VM Victory Score[j] =
Cloudlet Successful_j / Cloudlet Assigned_j
  End For
  Calculate Cluster Victory Score[i] = Sum
(Victory Score[j])
End For
Select OC = Max (Cluster Victory Score[i])
&& Cluste_Load[i] < Thrld
Return OC
```

Algorithm 1.2: Identify Resourceful Gateways
Input: Array of Fog Gateways N, Fog Threshold Load Thrld
Output: Array of Resourceful Gateways FogRes

```
   1. Identify_Resourceful_Gateways (N,
      Thrld)
   2. For n Ɛ N do
   3.   WorkLoad ← get.cpuload(n)
   4.   Memory ← get.memory(n)
   5.   Network ← get.network(n)
   6.   Energy ← get.energy(n)
   7.   If((Workload <= thrld) && (memory
      <= thrld) && (network <= thrld) &&
      (energy <= thrld)) then
   8.     FogRes[] ← n
   9.   End if
   10.End for
Return FogRes
```

Algorithm 1.3: Shuffle Prioritise Cloudlets
Input: Empty Priority Queue Array P, Cloudlet Array T
Output: Priority Queue Array P

```
   1. Shuffle_Prioritise _Cloudlets (T)
   2. For Task t Ɛ T do
   3.   p ← getpriority(t)
   4.   If (p != critical) then
   5.     PrioQueue[rear] ← task(t)
   6.   Else
   7.     PrioQueue[front] ← task(t)
   8.   End if
   9. End for
Return PrioQueue
```

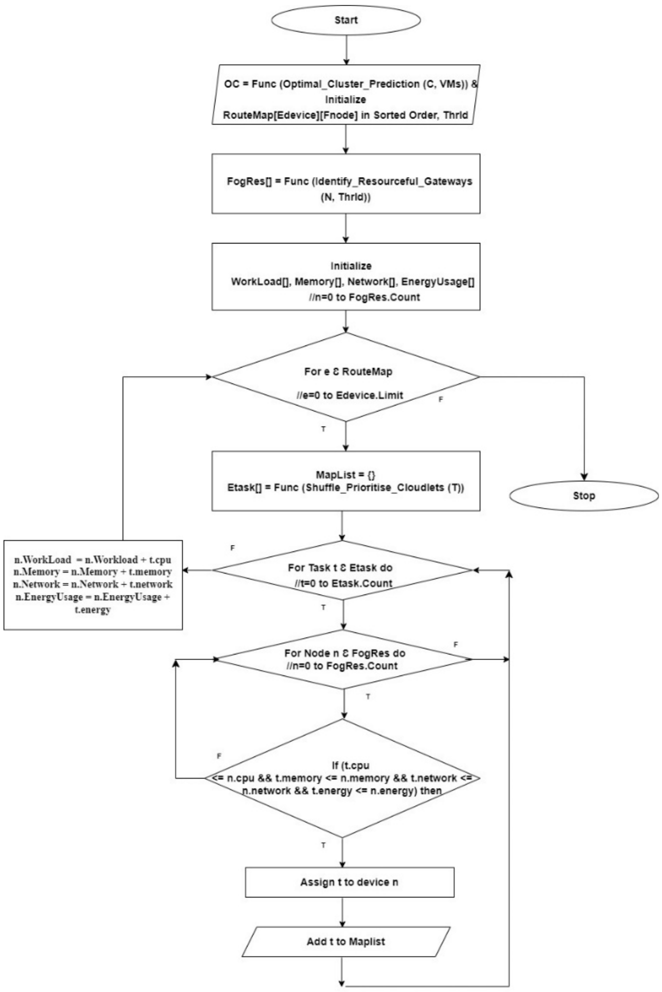

Fig. 3. Flowchart of the proposed EQLS.

As illustrated in Algorithm 1.1, each cluster's victory score is calculated based on the successful execution of the task request out of the total assigned request, and then, based on the global best among the cluster, the optimal cluster is selected as a fog gateway. As manifested in Algorithm 1.2, every computational node is evaluated to check whether the nodes have networking and processing resources available. Thus, based on this information, the system allocates tasks to the most suitable node. The incoming tasks from the end devices are sorted according to priority, as demonstrated in Algorithm 1.3, with high-priority tasks going to the front of the queue and low-priority tasks to the back. When a low-priority job is in the queue, the balancer determines whether there are any fresh high-priority tasks. If not, it first processes the low-priority functions according to the established policy. The priority of low-priority tasks is updated gradually after a period to avoid starvation. The flowchart at the end of this section that explains the proposed EQLS is depicted in Figure 3.

IV. EXPERIMENTAL PART

A. Simulation Setup

To illustrate the proposed strategy, the iFogSim [24] simulation tools are used on an AMD Ryzen 3, 2.60GHz CPU, 256GB SSD drive, and HP laptop running Windows 11 to execute the simulations of the integrated architecture as well as the surrounding environment. Considering that the fog has enough computing resources, the fog-based patient-centric smart healthcare system's performance is initially simulated concerning energy usage, response time, and cost. To replicate the environment on a big scale, a synthetic workload is employed in the simulations as the real-world workload for 100 to 1000 end users. The iFogSim Simulation Toolkit was utilized to implement the study problem of fog computing. The program was utilized to evaluate services and situations in a controlled environment. The iFogSim simulation engine was employed to fine-tune the system's fundamental performance concerns before deployment. iFogSim enables the simulation of infrastructure on a small or big scale, allowing the evaluation of alternative workload sets and resource performance. This allows adaptive application provisioning techniques to be more easily created, tested, and implemented. The network configuration of the proposed system deployed in the iFogSim simulator is shown in Table II. The fog devices are defined using CPU, memory, upward bandwidth, downward bandwidth, latency, and cost per MIPS.

TABLE II. CONFIGURATION OF THE PROPOSED SMART HEALTHCARE SYSTEM

Fog device	CPU (mips)	RAM (mb)	UP-BW (mbps)	DW-BW (mbps)	Latency (ms)	Cost/ Mips
Cloud	44800	30000	1000	1000	100	0.01
Proxy	22800	6000	1000	1000	10	0.03
Fog Node	11800	6000	1000	1000	5	0.06
Edge device	2800	3000	1000	1000	10	0.12

The physical topology of the proposed system, which consists of four master server nodes with two computing nodes attached to each master server node in the initial simulation, was extended up to 10 computing nodes per server node within a cluster as noticed in Figure 4.

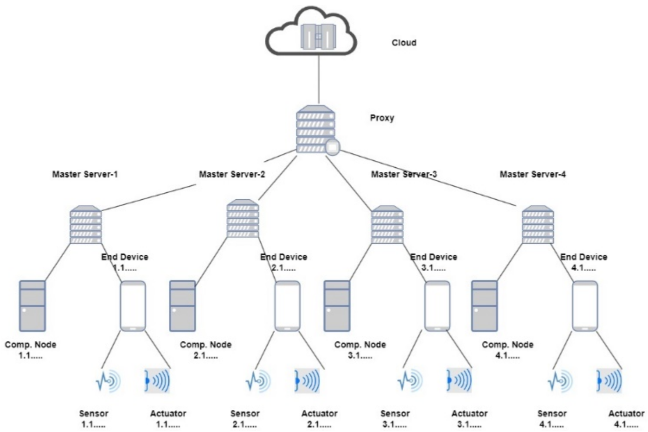

Fig. 4. Topology of the proposed smart healthcare system.

The first simulation setup with a fixed number of Computing Nodes (CNs) and different numbers of End Users (EUs) for analyzing the outcomes with different topologies is portrayed in **Table III**. Initially, a system simulation started with 100 EUs and 4 CNs within a cluster connected with the master server node.

TABLE III. SIMULATION SETUP WITH 4 CNs

Configuration	Setup-1	Setup-2	Setup-3	Setup-4	Setup-5
Cloud	1	1	1	1	1
Proxy	1	1	1	1	1
Server nodes	4	4	4	4	4
CNs	4	4	4	4	4
EUs	100	200	400	800	1000

The second simulation setup considered different CNs and a fixed number of EUs analyzing the outcomes of different topologies, as shown in Table IV. Initially, the system simulation started with 400 EUs and 2 CNs within a cluster connected with the master server node.

TABLE IV. SIMULATION SETUP 400 EUs

Configuration	Setup-1	Setup-2	Setup-3	Setup-4	Setup-5
Cloud	1	1	1	1	1
Proxy	1	1	1	1	1
Server nodes	4	4	4	4	4
CNs	2	4	6	8	10
EUs	400	400	400	400	400

B. Performance Estimation Metrics

A formulation explanation of the considered performance estimation metrics [25] follows.

1. **Response time (s or ms):** It represents the summation of all type of delays and the execution time of a task.

$$\text{Response\_time}_{ij} = \text{Latency}_{ij}^{cm} + \text{Exeution\_time}_{ij} \quad (1)$$

2. **Energy Usage (kJ):** It represents the overall power consumed by the resources to execute the jobs.

$$\text{Energy\_Usage}_{ij} = \sum_{i=1}^{n} \text{Power}_{\text{Trans (i)}} + \text{Power}_{\text{exec (i)}} + \text{Power}_{\text{Sense (i)}} \quad (2)$$

3. **Cost ($):** The overall cost incurring while executing the tasks using the selected resources.

$$\text{Cost}_{\text{Comp}} = \sum_{i=1}^{n} \text{Agent}_{\text{cpumips}_i} \text{ X Agent}_{\text{exectime}_i} \text{ X Agent}_{\text{ratemips}_i} \quad (3)$$

V. RESULTS AND DISCUSSION

This section provides the EQLS performance evaluation findings compared to the FNPA and the RR schemes. Regarding response time and energy use, EQLS reduces both and produces ideal results compared to the FNPA and the RR. Experimental results for the performance parameters of response time and energy use indicate that the proposed fog-based system is a good choice for smart healthcare systems when rapid data processing is vital. The simulations were conducted 100 times per setup to record the outcomes of remotely monitoring patients using the patients' medical entries.

The patient uses body-worn sensors to collect the data. The patient's smartphone transmits the data that the sensors have collected to the diagnostic module to be processed.

The first simulation results of EQLS with a fixed number of computing nodes (4) and a growing number of end users (100, 200, 400, 800, 1000), as defined in Table III**,** are presented in Table V. It can be inferred that for a certain number of fog nodes connected to the cluster controller node, the related costs and response time increase as the number of IoT devices increases due to higher computational and communication delay. However, there is a minor increase in energy consumption due to the effective load balancing of task requests among the fog nodes originating from IoT devices. Adding more fog nodes per server in a cluster with a rising number of IoT/end devices, may enhance the outcome by utilizing the proposed method.

TABLE V. SIMULATION RESULTS OF EQLS FOR VARYING NUMBER OF EUs

Using the setup of Table III			
Config.	**Response time (s)**	**Energy usage (kJ)**	**Cost ($)**
Setup-1	0.94	169.31	2155.67
Setup-2	1.69	172.75	5691.73
Setup-3	2.47	174.31	7478.13
Setup-4	4.89	179.70	9316.34
Setup-5	6.56	183.00	9445.63

The proposed EQLS approach has a 6.2% and 13.4% faster response time than the FNPA and the RR, respectively, as can be seen in Figure 5. Similarly, the former approach used 8.7% and 14.9% less energy than the FNPA and the RR schemes, respectively, as exhibited in Figure 6. EQLS, costs 7.6% more than the FNPA but 13.7% less than the RR scheme, as observed in Figure 7.

The results of the second simulation group, for the proposed EQLS with different CNs (2, 4, 6, 8, 10) and a fixed number of EUs (400), as defined in Table IV, for analyzing the outcomes with different topologies, are portrayed in Table VI. It can be seen that for a certain number of IoT devices, the related costs and response time decrease as the number of fog nodes linked to the cluster controller node increases up to a certain point. Following this, owing to increasing computational and communication delay, both parameter values begin to rise after that specific point. However, due to the effective load balancing of task processing requests across the cluster, fog nodes originating from IoT/End devices manifest a reduction in energy consumption until it stabilizes.

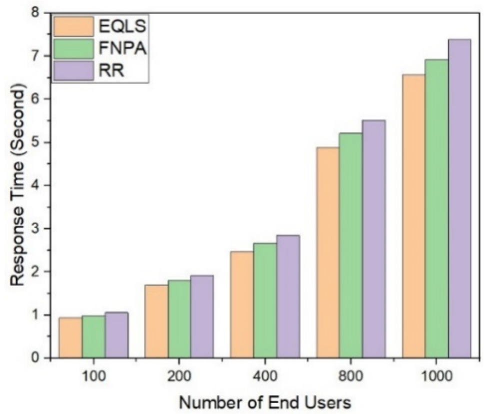

Fig. 5. Comparative results considering response time for 4 CNs.

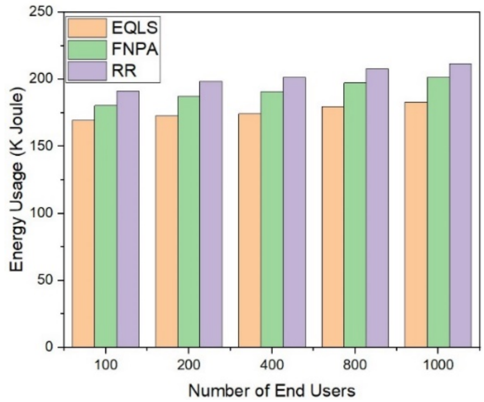

Fig. 6. Comparative results considering energy usage for 4 CNs.

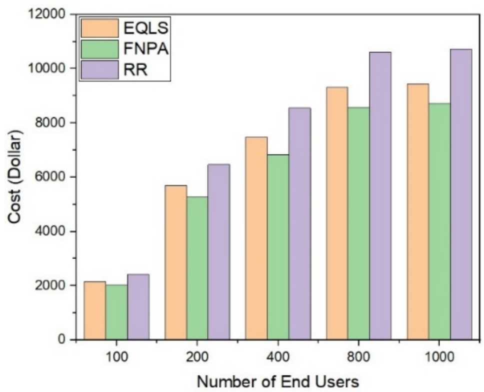

Fig. 7. Comparative results considering cost for 4 CNs.

The response time plotted against the number of computing nodes in a cluster is depicted in Figure 8. It shows that the response time drops significantly up to a certain point of increase in CNs, and then it starts to climb again due to the increasing communication latency. As the number of CNs in a cluster increases, the energy consumption decreases after a specific level, as illustrated in Figure 9, due to additional fog nodes which are available to handle task requests. The cost displays a progressive drop up to a specific limit of CNs within a cluster and then rises due to the increased computational delay, as exhibited in Figure 10.

TABLE VI. SIMULATION RESULTS OF EQLS FOR VARYING CNs

Using the setup of Table IV			
Config.	**Response time (s)**	**Energy usage (kJ)**	**Cost ($)**
Setup-1	2.60	182.75	7478.63
Setup-2	2.54	174.08	7404.24
Setup-3	2.43	171.62	7295.37
Setup-4	2.55	170.63	7324.96
Setup-5	2.62	169.80	7464.13

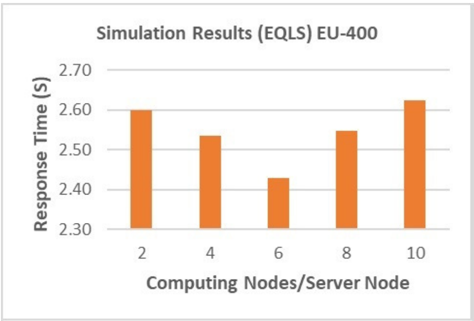

Fig. 8. Comparative results considering response time for 400 EUs.

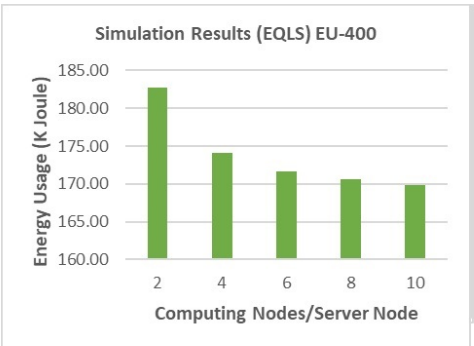

Fig. 9. Comparative results considering energy usage for 400 EUs.

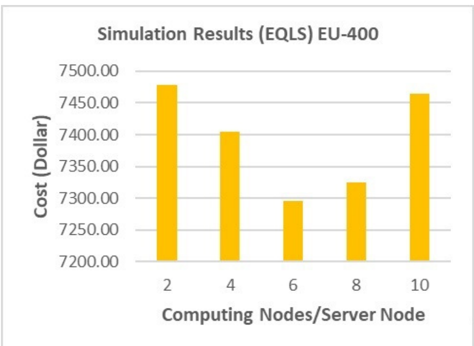

Fig. 10. Comparative results considering cost for 400 EUs.

In summary, the simulation demonstrated considerable reductions in response time and energy consumption with a cost penalty as the computation cost is higher on fog nodes. With the proposed scheme, most of the computations will now be possible on fog gateways. The first main point emerging from this paper is that a delay-sensitive smart healthcare system would greatly benefit from the introduced system. Regarding performance in comparison to two current load-balancing approaches, FNPA and RR, the results favor the proposed EQLS approach.

VI. CONCLUSION AND FUTURE SCOPE

In smart cities, the most significant healthcare cost benefits from cloud computing come from moving services to the patient's home. Cloud services can help the early diagnosis and treatment of health problems. However, programs that require precise timing may not be a good fit for cloud computing. Fog services can improve the scalability, data processing, and dependability of IoT-based healthcare systems in smart cities. Fog computing offers flexible and lightweight resources to patient devices in healthcare systems. Edge computing routers, switches, and constrained computing devices can carry out some tasks closer to the source using fog services. This lowers latency, increases service flexibility, distributes resource requirements, and avoids multi-hop data transfer. To address current problems in smart healthcare systems, this paper proposes a fog-driven patient-centric smart healthcare system for the e-healthcare environment of smart cities. Time-sensitive real-time smart healthcare applications are benefited from stabilizing fog node burden through the proposed Multi-objective Energy-efficient QoS-aware Load balancing Strategy (EQLS). Employing iFogSim simulations, patients' cardiac condition is evaluated remotely. The simulation outputs are compared to those of existing load-balancing policies (Round Robin (RR) and Fog Node Placement Algorithm (FNPA)) regarding cost, response time, and energy usage. According to the experimental data, when compared to the FNPA policy, the proposed technique decreased response time by 6.2% and energy usage by 8.7%. The proposed strategy also reduced response time by 13.4% and energy usage by 14.9% in comparison with the RR policy.

The introduced system addresses the crucial requirement for multi-objective optimization. This strategy effectively manages the trade-off between response time and energy consumption to achieve the best outcomes. It considers job priorities and selects the optimal cluster fog resources for executing task requests. The proposed fog-driven patient-centric smart healthcare system can be extended for further research in cluster intelligence to minimize cost, module sharing, and patient mobility under different application scenarios.

REFERENCES

[1] A. El Azzaoui, S. K. Singh, and J. H. Park, "SNS Big Data Analysis Framework for COVID-19 Outbreak Prediction in Smart Healthy City," *Sustainable Cities and Society*, vol. 71, Aug. 2021, Art. no. 102993, https://doi.org/10.1016/j.scs.2021.102993.

[2] S. K. Singh, Y. Pan, and J. H. Park, "Blockchain-enabled Secure Framework for Energy-Efficient Smart Parking in Sustainable City

Environment," *Sustainable Cities and Society*, vol. 76, Jan. 2022, Art. no. 103364, https://doi.org/10.1016/j.scs.2021.103364.

[3] M. Hamdani, M. Youcefi, A. Rabehi, B. Nail, and A. Douara, "Design and Implementation of a Medical TeleMonitoring System based on IoT," *Engineering, Technology & Applied Science Research*, vol. 12, no. 4, pp. 8949–8953, Aug. 2022, https://doi.org/10.48084/etasr.5040.

[4] M. Aknan, M. P. Singh, and R. Arya, "AI and Blockchain Assisted Framework for Offloading and Resource Allocation in Fog Computing," *Journal of Grid Computing*, vol. 21, no. 4, Nov. 2023, Art. no. 74, https://doi.org/10.1007/s10723-023-09694-7.

[5] S. Singhal, S. Betgeri, and S. K. Singh, "Strategies for Mitigating Security Concerns in IoT-Enabled Smart Cities," in *Secure and Intelligent IoT-Enabled Smart Cities*, Hershey, PA, USA: IGI Global, 2024, pp. 239–273.

[6] K. N. Tun and A. M. Myat Paing, "Resource Aware Placement of IoT Devices in Fog Computing," in *International Conference on Advanced Information Technologies*, Yangon, Myanmar, Nov. 2020, pp. 176–181, https://doi.org/10.1109/ICAIT51105.2020.9261787.

[7] T. Akhtar, N. G. Haider, and S. M. Khan, "A Comparative Study of the Application of Glowworm Swarm Optimization Algorithm with other Nature-Inspired Algorithms in the Network Load Balancing Problem," *Engineering, Technology & Applied Science Research*, vol. 12, no. 4, pp. 8777–8784, Aug. 2022, https://doi.org/10.48084/etasr.4999.

[8] S. Pallewatta, V. Kostakos, and R. Buyya, "Microservices-based IoT Application Placement within Heterogeneous and Resource Constrained Fog Computing Environments," in *12th IEEE/ACM International Conference on Utility and Cloud Computing*, Auckland, New Zealand, Dec. 2019, pp. 71–81, https://doi.org/10.1145/3344341.3368800.

[9] M. Goudarzi, H. Wu, M. Palaniswami, and R. Buyya, "An Application Placement Technique for Concurrent IoT Applications in Edge and Fog Computing Environments," *IEEE Transactions on Mobile Computing*, vol. 20, no. 4, pp. 1298–1311, Apr. 2021, https://doi.org/10.1109/TMC.2020.2967041.

[10] A. M. Rahmani *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, pp. 641–658, Jan. 2018, https://doi.org/10.1016/j.future.2017.02.014.

[11] O. Akrivopoulos, I. Chatzigiannakis, C. Tselios, and A. Antoniou, "On the Deployment of Healthcare Applications over Fog Computing Infrastructure," in *41st Annual Computer Software and Applications Conference*, Turin, Italy, Jul. 2017, vol. 2, pp. 288–293, https://doi.org/10.1109/COMPSAC.2017.178.

[12] P. G. V. Naranjo, Z. Pooranian, M. Shojafar, M. Conti, and R. Buyya, "FOCAN: A Fog-supported smart city network architecture for management of applications in the Internet of Everything environments," *Journal of Parallel and Distributed Computing*, vol. 132, pp. 274–283, Oct. 2019, https://doi.org/10.1016/j.jpdc.2018.07.003.

[13] R. Mahmud, F. L. Koch, and R. Buyya, "Cloud-Fog Interoperability in IoT-enabled Healthcare Solutions," in *19th International Conference on Distributed Computing and Networking*, Varanasi, India, Jan. 2018, pp. 1–10, https://doi.org/10.1145/3154273.3154347.

[14] 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," *Computers & Electrical Engineering*, vol. 67, pp. 58–69, Apr. 2018, https://doi.org/10.1016/j.compeleceng.2018.02.047.

[15] M. Al-khafajiy, L. Webster, T. Baker, and A. Waraich, "Towards fog driven IoT healthcare: challenges and framework of fog computing in healthcare," in *2nd International Conference on Future Networks and Distributed Systems*, Amman, Jordan, Jun. 2018, pp. 1–7, https://doi.org/10.1145/3231053.3231062.

[16] 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, Nov. 2018, https://doi.org/10.1016/j.compeleceng.2018.08.015.

[17] S. S. Gill, R. C. Arya, G. S. Wander, and R. Buyya, "Fog-Based Smart Healthcare as a Big Data and Cloud Service for Heart Patients Using IoT," in *International Conference on Intelligent Data Communication Technologies and Internet of Things*, Coimbatore, India, Aug. 2018, pp. 1376–1383, https://doi.org/10.1007/978-3-030-03146-6_161.

[18] S. Tuli *et al.*, "HealthFog: An ensemble deep learning based Smart Healthcare System for Automatic Diagnosis of Heart Diseases in integrated IoT and fog computing environments," *Future Generation Computer Systems*, vol. 104, pp. 187–200, Mar. 2020, https://doi.org/10.1016/j.future.2019.10.043.

[19] S. Malik *et al.*, "Intelligent Load-Balancing Framework for Fog-Enabled Communication in Healthcare," *Electronics*, vol. 11, no. 4, Jan. 2022, Art. no. 566, https://doi.org/10.3390/electronics11040566.

[20] B. Premalatha and P. Prakasam, "Optimal Energy-efficient Resource Allocation and Fault Tolerance scheme for task offloading in IoT-FoG Computing Networks," *Computer Networks*, vol. 238, Jan. 2024, Art. no. 110080, https://doi.org/10.1016/j.comnet.2023.110080.

[21] D. Deepa and K. R. Jothi, "Classification of Request-Based Mobility Load Balancing in Fog Computing," *Computer Systems Science and Engineering*, vol. 46, no. 1, pp. 137–151, 2023, https://doi.org/10.32604/csse.2023.032485.

[22] M. Talaat, A. Saleh, M. Moawad, and J. Zaki, "Fog computing effective load balancing and strategy for deadlock prediction management," *Ain Shams Engineering Journal*, vol. 14, no. 12, Dec. 2023, Art. no. 102561, https://doi.org/10.1016/j.asej.2023.102561.

[23] Y. Wang, W. Shafik, J.-T. Seong, A. Al Mutairi, M. SidAhmed Mustafa, and M. R. Mouhamed, "Service delay and optimization of the energy efficiency of a system in fog-enabled smart cities," *Alexandria Engineering Journal*, vol. 84, pp. 112–125, Dec. 2023, https://doi.org/10.1016/j.aej.2023.10.034.

[24] H. Gupta, A. Vahid 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," *Software: Practice and Experience*, vol. 47, no. 9, pp. 1275–1296, 2017, https://doi.org/10.1002/spe.2509.

[25] A. U. Rehman *et al.*, "Dynamic Energy Efficient Resource Allocation Strategy for Load Balancing in Fog Environment," *IEEE Access*, vol. 8, pp. 199829–199839, 2020, https://doi.org/10.1109/ACCESS.2020.3035181.

AUTHORS PROFILE

Gaurav Goel received his B.Tech in IT from UPTU, M.E. in CSE as a Gold Medalist from Panjab University and is pursuing a PhD from BTU. Mr. Goel has a total of 16 years of teaching and industry experience. He currently serves as the youngest associate dean of MIET, a 25+ years old institute of AKTU. He has authored many articles in national and international conferences and journals. His research interests include image processing, fog computing, and cloud computing.

Amit K. Chaturvedi is an Assistant Professor at the Government Engineering College, Ajmer. He earned a M.Sc. from M.P. Bhoj University, Bhopal, and a PhD from Bhagwant University, Ajmer. He has published 65+ articles in reputable national and international journals and four chapters in edited books. He has presented multiple research articles at indexed national and international seminars, conferences, and workshops. He has also organized several national and international seminars and workshops. He has made significant research contributions in adhoc networks and cloud-fog computing.