

Research Article

A Personalized Eccentric Cyber-Physical System Architecture for Smart Healthcare

Amutha Balakrishnan,¹ Ramana Kadiyala ,² Gaurav Dhiman ,³ Gokul Ashok,¹ Sandeep Kautish ,⁴ Kusum Yadav,⁵ and J. Maruthi Nagendra Prasad ⁶

¹*School of Computing, SRM Institute of Science and Technology, Chennai, India*

²*Department of Artificial Intelligence & Data Science, Annamacharya Institute of Technology and Sciences, Rajampet, Andhra Pradesh, India*

³*Department of Computer Science, Government Bikram College of Commerce, Patiala, India*

⁴*Dean-Academics with LBEF Campus, Kathmandu, Nepal*

⁵*College of Computer Science and Engineering, University of Ha'il, Ha'il, Saudi Arabia*

⁶*Department of Computer Science and Engineering, Annamacharya Institute of Technology and Sciences, Rajampet, Andhra Pradesh, India*

Correspondence should be addressed to Sandeep Kautish; sandeep.kautish@lbef.edu.np

Received 27 September 2021; Revised 28 October 2021; Accepted 2 November 2021; Published 17 December 2021

Academic Editor: Thippa Reddy G

Copyright © 2021 Amutha Balakrishnan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The development and technological advancement of wireless sensor networks in different fields has been a revolution for mankind. To meet the high-end requirements, the support of the cloud that provides the resources for the application is very much essential. This paper presents an architecture called cloud sense to connect cyber and physical spaces for wireless body area networks with varying high-end workflow at different perspectives. The scalability issue in collecting patient data and processing the data is established using ganglia that is a scalable, distributed monitoring system to support high-performance computing in clusters for the set of input events such as electrocardiogram (ECG), blood pressure (BP), saturation of peripheral oxygen (SPO₂), temperature, and skin conductance of the kind of human body parameters. Various parameter metrics have been analyzed based on the equivalent creation of instances. The connectivity mechanism behind the proposed cyber-physical system is unique of its kind; it is exhibited through wireless Internet on a small scale of three remote locations; the system works well with specific network parameter metrics; and the results proved that availability and scalability issues were addressed with numerical analysis.

1. Introduction

A cyber-physical system (CPS) is required to interconnect the physical devices in the hospital for healthcare monitoring and to analyze the data stored in the cloud. Besides, an analytics platform through the internet of things (IoT) is the need of the hour for efficient healthcare delivery in the world. The proposed CPS will act as an interface between the physical and cyber worlds. Physical world comprises the body sensors and the electronic devices that can be interconnected together to form the physical space. Cyberspace consists of the data where it can be transferred to the doctors

and the researchers to analyze and make decisions for further need. The CPS that is proposed using the smart health application was operational under the following categories: (1) patient-centric, (2) network-centric, (3) hardware-centric, and (4) data-centric. CPS will function as an intelligent monitoring platform for timely diagnostic decisions in the critical care unit of hospitals and home care patients. This research lies in blending recent cutting-edge technologies that are wireless body area network (WBAN), IoT, and streaming big data analytics to handle an enormous amount of data. The concept of fog computing has been introduced to save energy and time to provide timely and

needy services at the doorstep of the patient. Also, an algorithm for disseminating the patients' health parameters using a priority queuing mechanism is proposed. By introducing fog, data analytics is possible in the terminals themselves, which again improves adeptness in the proper functioning of the system with localized decision-making, the geographical distribution of data within less time, and optimized usage of resources. Fog computing enables people to collect data from various devices and has a larger capacity to process more data than edge computing, whereas edge computing performs much of the processing on embedded computing platforms kept with the patients in a WBAN system as it is directly interfaced with the sensors and controllers.

The proposed system will reduce latency, improve operational efficiency, and will provide effective service to save human life using built-in decision-making policies by the introduction of fog controllers, which are used for effective data dissemination locally with reduced time complexity.

1.1. Problem Definition. The research tries to interconnect the recent cutting-edge technologies that are wireless body area network, internet of things (IoT), and streaming big data analytics. The concept of fog computing has been introduced to save energy and time to provide timely and needy services nearer to the doorstep of the patient regardless of the location of the patient. Hospitals, doctors, and patients are interconnected through local and remote servers, through the fog controllers into the cloud. Also, an algorithm for disseminating the patients' health parameters using a priority queuing mechanism is proposed. By introducing fog, data analytics is possible in the end terminal itself, which again improves adeptness in the proper functioning of the system with localized decision-making, the geographical distribution of data in lesser time with optimized usage of resources. When a massive amount of data needs decision-making, scalability and reliability issues have been solved by the concept of availability.

2. Literature Review

Future global deployment of WSNs could provide data in petabytes or exabytes every year for, for example, environmental monitoring. Whether the related cloud environment model is appropriate for the processing of sensor information is, however, far from clear [1]. The next-generation network sensor platforms should aim for a multi-application model of popular infrastructure with a strong separation of concerns between infrastructure providers and application developers. The WBAN ecosystem can be applied to the cloud, and infrastructure as a service (IaaS) providers such as Amazon EC2 or Eucalyptus can provide an infrastructure for healthcare [2]. The WBAN networks provide a way to capture physiological data for use in several distributed applications. To provide end-to-end physiologic monitoring and diagnosis [3], Amazon EC2 can respond to the complex needs of medical services and can be incorporated with wireless body sensor network technologies.

However, the large amount of data stored in the cloud is easily measured for performance in real time. MapReduce is the most prevalent cloud computing programming model [2]. It is the programming model for the processing and generation of large data sets. In the MapReduce model, several real-world activities are expressible. Functional-style programmes are automatically paralleled and run on a wide variety of commodity machines. The run-time framework provides descriptions of the partitioning of input data, coordinating the execution of the programme across several machines, handling machine failures, and maintaining contact between the machines [3]. The WBAN data can be fit into a model by updating the model parameters that enables two or more databases to appear as one, whether on premises or in the cloud. Teradata QueryGrid, IBM PureData Systems with Fluid Query, and SAP HANA that work with smart data services are offering data federation capabilities. The model itself is sufficient for the physicians to go for a decision without affecting the underlying WBAN data.

Previous research in the health sector centered on creating the prototype of the body area network with wireless sensors for the use of routing protocols. Nodes in the body sensor were used to detect critical human parameters such as ECG, blood pressure, level of oxygen, heart rate, and body temperature. The paper "Alerts for mobile healthcare: criteria and pilot studies" provides efficient routing and tracking of alerts to quality and cost-effective health facilities [4]. In their paper, Lee et al. proposed that high blood pressure and arrhythmia can be effectively avoided and regulated by continuous physiological surveillance [5]. Previously, a smart, mobile care system focused on roles with an alerting mechanism was proposed and implemented [2]. For further study, variations of human body parameter values in various patients are reported when standard human body parameters are retained as median values. An algorithm has been developed to classify human values anomalies leading to the diagnosis of disease and medicines [6]. Several values of human body parameters were collected and translated into unique data packets for doctoral evaluation and wirelessly forwarded to a hospital server. When the observed human body parameters were greater than the threshold values, an alert message was suggested for the caregiver assigned to the patient with encrypted contact [7]. The ZigBee system for fall tracking, incorporated through drop detection, indoor positioning, and ECG monitoring, offered insights into a secure transmission protocol based on anycast routing for the wireless patient surveillance process [8]. WBAN writers performed a report on medical and nonmedical uses. It offers a great deal of insight into the applications. The medium access protocol has been revised to collect patient data for context awareness purposes [9]. The authors in [10] proposed a knowledge interview mechanism for globally accessing patient data that was clarified in the cloud-based wireless body sensor network. In [11], the authors made a profound survey study of wireless body area network architecture problems.

In [12], the authors suggested a fuzzy logic application to diagnosis of anemia for expert fuzzy system presentation. It

has an expert system based on an inflammatory system that was only considered for the diagnosis of amnesia and other body parameters. The authors discussed a 1 kg rise in body weight, correlated with the systemic increase in blood pressure between 3 and 6 mm Hg in the healthy and ingravescens classes [13]. Soon, telehealth initiatives are also underway. Researchers also discussed the role of an intelligent mobile care system with a warning mechanism in the chronic care climate. The device tracks patients' medical records and sends rapid avoidance measures to avoid unexpected promises [14]. A structure for collaborative software agents has been presented by separate software agents with three main components: information management, the reasoning for confusion, and software agents [15]. The framework for the structured handling of warning messages was created as an alert monitor that complies with the needs of the medical staff or its mobile devices for receiving warnings within a specific deadline [16].

The authors' findings suggested that the cloud-integrated sensor data provides a special hybrid platform for remote health surveillance [5]. The goal was very well defined to provide valuable insight for the designers of WBANs and highlight key problems concerning the efficiency of the collection of healthcare data [17]. An overview of the computer environment has been studied for the collection of personal data from remote mobile patients [18]. The authors proposed a smart health solution through the use of a clustering mechanism for wireless sensor networks [19]. Mobile ZigBee and Bluetooth health gateways have been briefly analyzed [20]. An integrated gateway for various PHDs was implemented to collect measurements from different PHDs. It functions in two modes, namely, immediate transmission and integrated transmission. The overhead transmission can be minimized by the gateway consisting of an activity monitor, a drug dispenser, and a pulse oximeter [21]. For remote patient monitoring applications as a trial, an intelligent smart health portal with fog was initialized [22].

As IoT leads to an exponential proliferation of endpoint systems, fog computing is known to expand the hierarchically distributed architecture from the edge of the network to the heart. In addition to big data and analytics, IoT introduces a new dimension to its wide distribution of sources [23]. In the field of healthcare applications, the basic computing materials of fog were treated where data can be moved without delay [24]. The technology acceptance model (TAM) has been developed and has shown the difference in health conditions between adoption factors because of the advancement of medical technologies and their perceived ease of use [25]. A research by Megalingam et al. [18] suggested a portable system to give warnings to the caretakers. The researchers addressed the development of virtual group enablers (VGE) between patient, nurse, and doctor devices to allow the remote analysis of WBAN data. The study involves GMS, the medical data recording server (MDRS), the policy engine (PE) and medical officers' equipment, WBANs for patients, and environmental sensors. Group preparation and adjustment have been undertaken depending on the circumstances and needs of

patients and medical officers, which can be easily modified by high-level policies. Medical officers provide input on the consistency of the obtained WBAN data by using quality of health monitoring [26]. Authors have suggested a secure transmission protocol based on all cast routing for wireless patient monitoring, which automatically selects the nearest recipient data in anycast category to minimize millisecond latency and control overhead [27]. The above literature did not concentrate on dispersed needy service without latency at the appropriate geographical locations, whereas the proposed project focuses mainly on managing loads by utilizing resources properly, providing geographically distributed customer needs with minimal latency and maximizing resource usage from nearest points of interest, during the sequence.

So there is a need for a scalable architecture for healthcare as a case study with many numbers of instances created using virtual machines in a cloud computing environment to interconnect patients, doctors, and hospitals geographically. The paper is organized as follows. Section 3 represents the five-tier methodology and subsection details. Section 4 reveals the cyber-physical system components. Section 5 presents the mathematical modeling of five vital parameters of WBAN. Section 7 presents the concluding remarks with a case study after Section 6 with performance metrics.

3. Proposed System

The proposed system will reduce latency, improve operational efficiency, and will provide effective service to save human life using built-in decision-making policies in the observed WBAN data. Remote healthcare through fog computing is one of the new approaches that can handle some of the challenges of smart healthcare in terms of localized decision-making, geographical distribution of data, and smart load balancing with security, sharing, integration, and management. In the proposed work, we implemented the significance and opportunities of fog essentials to reduce the tasks offered by cloud computing is pervasive in healthcare's future challenges it faces as of today. The proposed architecture consists of five tiers as shown in Figure 1. The methodology in the context of tier-wise software, hardware, and the proposed algorithms is described in the consecutive flow charts. Figure 1 indicates the proposed architecture to fuse cyber and physical phases in the domain of wireless healthcare in terms of five tiers. When the computational needs of individual tasks are high, the workflow is categorized as calculation-intensive. Similarly, when data specifications are fantastic (e.g., size of and data file, number of files, data storage, etc.), the workflow is categorized as data-intensive. Data-intensive workflows may use the architecture of environments such as data clouds. Data clouds offer services such as low-latency transportation protocols and reproduction mechanisms for data delivery, for which massive data sets stored in distributed repositories need to be accessed, processed, and transmitted. The period depends on the time you spend dealing with the input and output files and the time you compute them.

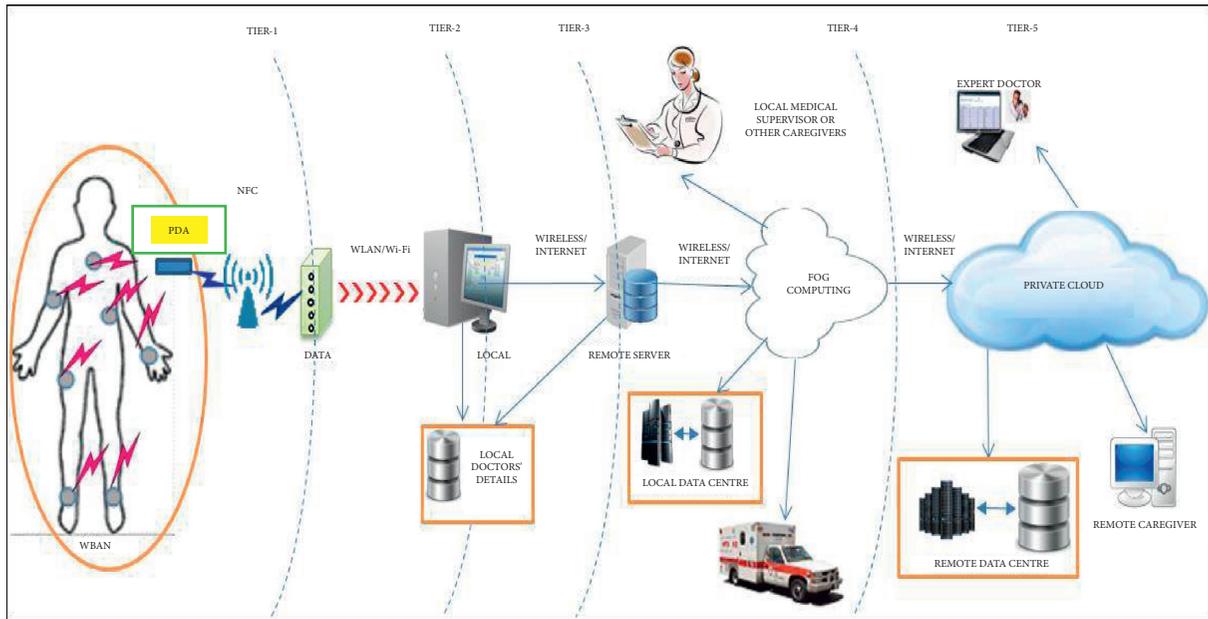


FIGURE 1: Proposed five-tier architecture for remote medical care fusing cyber and physical spaces.

In a traditional healthcare monitoring system, we rely on human intervention along with physical devices for observing patient information periodically and consistently. The internet of things (IoT) is a connecting system, such as electronic devices, buildings, and even more medical centers and hospitals, in which full access to all patient information is communicated at needy times with assured data through the Internet. To assist in the healthcare modernization process, WBAN along with IoT devices connected to the Internet plays a major role. The human healthcare framework is a patient-resource-based monitoring system that includes informational, audiovisual coordination, and the retrieval of health data through the sensors. The integration of IoT in healthcare monitoring systems is cumbersome because of the large amounts of data and the need for secured data transfer to protect patient's personal medical information from being seeped [28]. Any malicious person's intervention or stealing of data and altering or modifying the patient data for any unwanted and unethical purposes lead to data violation, and manipulation of patient's vital data in any way may have serious implications; even it may lead to the death of the patient [29]. Therefore, building a definite architecture for a wireless body area network with varying workflow is considered with location information as an application domain. The scalability problem is addressed by suitable algorithms when data flow becomes enormous. The virtualization concept is applied to address when a large handling issue occurs in the cloud environment.

3.1. Tier I: Data Collection from WBAN to PDA. The medical data collected from the body sensor network comprises ECG, SPO_2 , pulse rate, temperature, skin conductance, and blood pressure. A data collector that is built within the microcontroller unit collects all the six-sensor data and processes the data based on prioritization. The body sensor

data is prioritized as normal, abnormal, and critical. A personal digital assistant (PDA) is responsible for collecting the data from the sensors using near-field communication (NFC). NFC, Bluetooth, and ZigBee are provided with information on chip vendors and application product vendors' deployment in smart healthcare services. Based on the availability, we can go for the communicating device. The algorithm resides in the controller to check the incoming data with a normal human body data set. If there is no data from the sensors for a specified period of an interval, the loop re-executes to access the data from the body sensors with a significant waiting time as represented in Figure 2.

3.2. Tier II: Routing Data from PDA to Local Server. The PDA checks the integrity of the incoming data with the previous data. The data collected from the respective PDAs was routed to the local server. The PDA ID and the doctor ID are mapped according to the specialization, using a local doctor's database available in the local server. The streaming analytics engine continuously monitors the parameters for critical and abnormal patients. The summary of parameters was sent to the local doctor and the emergency response team in the hospital. The mapping is stored and could be retrieved through a web page, using the categorized status of normal, abnormal, and critical as represented in Figure 3.

3.3. Tier III: Routing Data from Local Server to Remote Server. The mapped information was routed from the local servers at a different location to the specified remote server. The remote server has a database that consists of a cluster of expert physicians to be referred across different hospitals and geographical locations for needy patients. It works on a cluster binding algorithm, which binds the patient information to the appropriate doctor from the cluster while

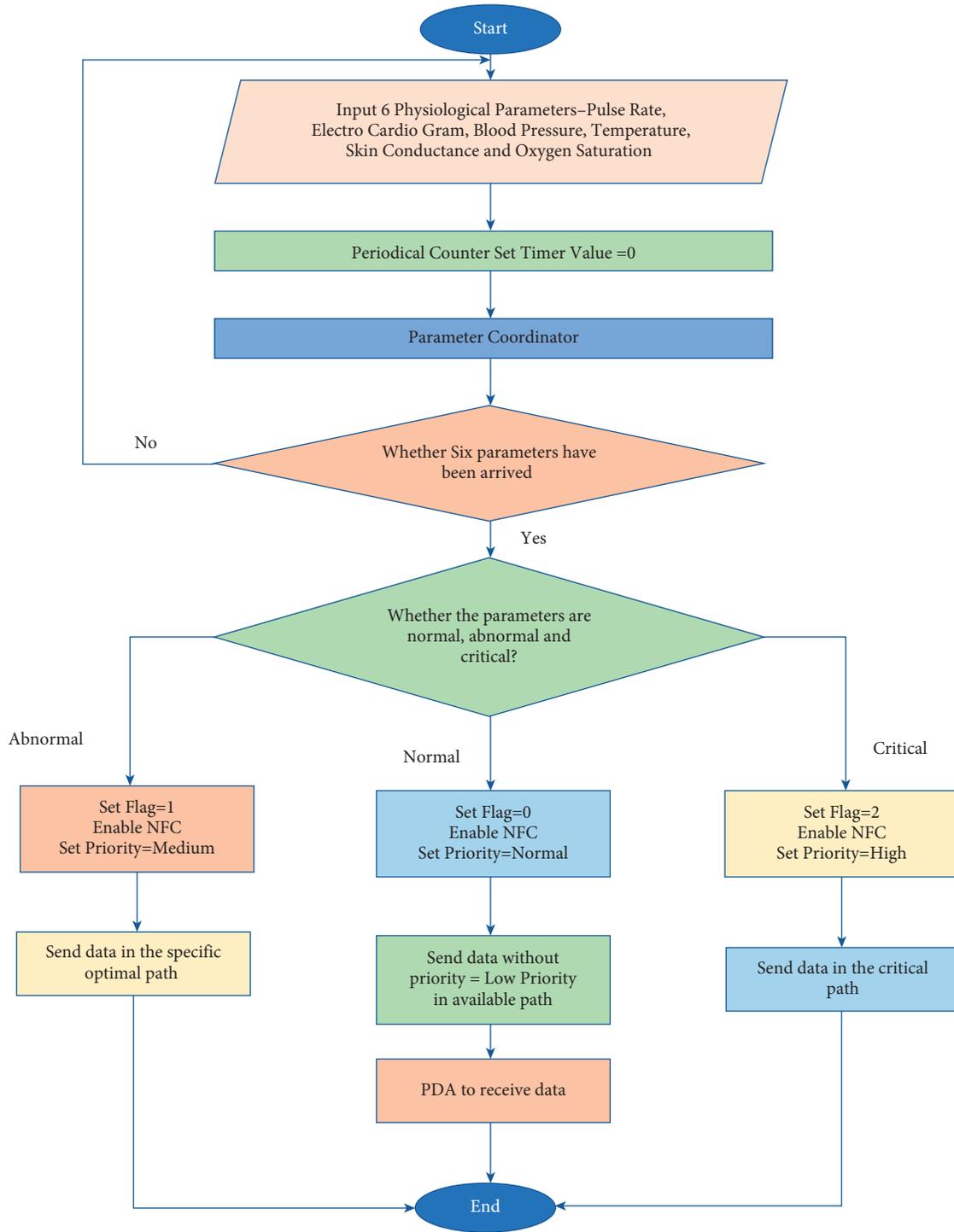


FIGURE 2: TIER I: from medical kit to PDA; mode of communication: NFC.

operating and gathering the data. This is an improved system where clustering has been centralized for collecting and managing data depending on the network parameters and the availability of appropriate doctors. This acts as an aid in the remote server to view the mapped information. The previous mapping of PDA with a doctor in Tier II is done for doctors in the same location whereas the binding in the remote server does the mapping with an additional doctor in the nearest neighboring locations as represented in Figure 4.

Figure 2 emphasizes collecting the six physiological parameters and verifies whether all the six vital parameters of the patient have been received. An abnormality table is constructed in such a way that, whether they received six vital parameters fall within the threshold, say a safer health status, assuming normal and if the parameters go behind the threshold slightly lesser or higher, it can be treated as abnormal and if the parameters shoot up and show turbulence and are also lesser or higher beyond the threshold, it is

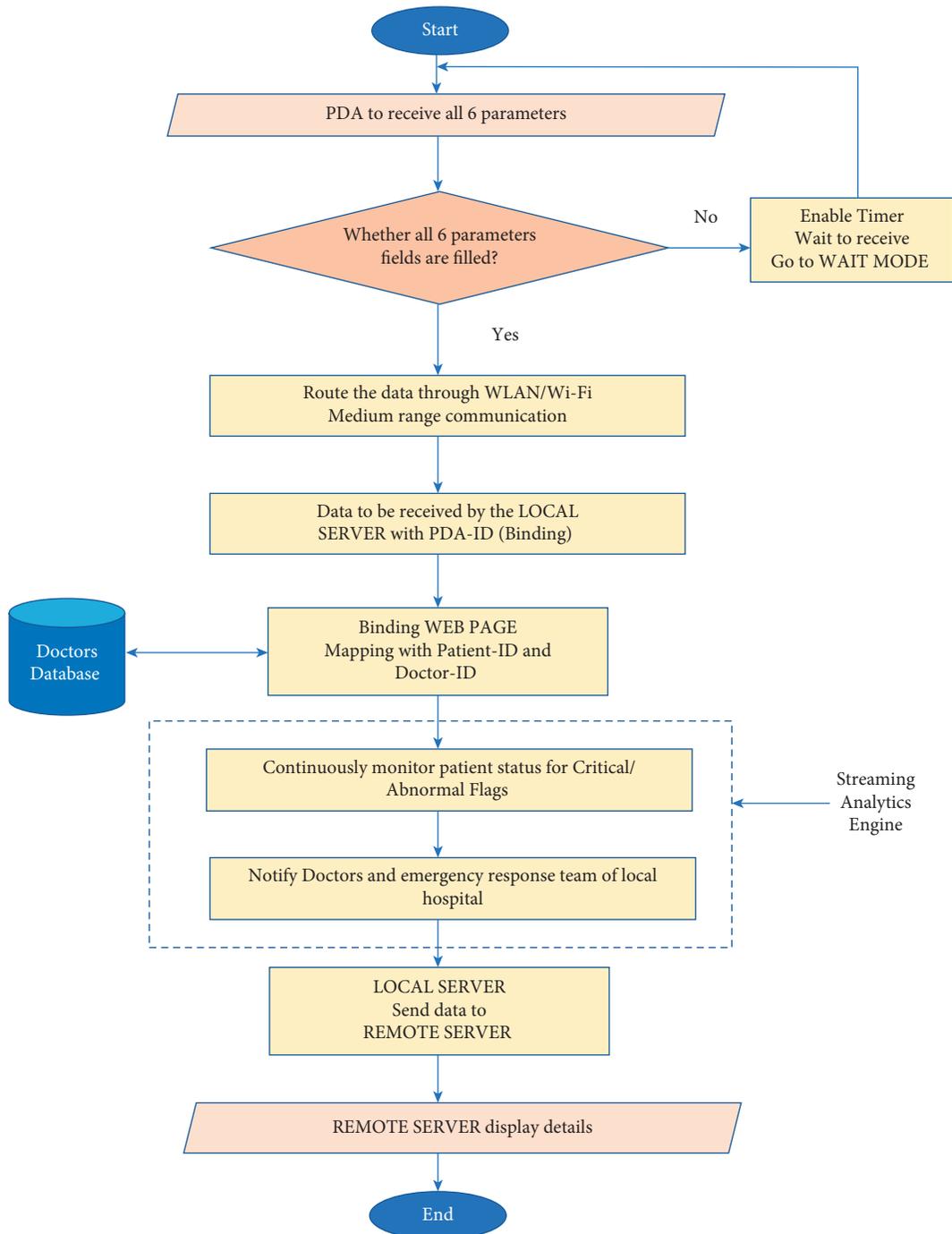


FIGURE 3: TIER II: from PDA to local server; mode of communication: wireless internet.

subsumed as critical after the comparison of the benchmarked data set.

Figure 3 prioritizes the routing of data after receiving from the PDA, and a mapping is done with respect to patient and physician. A streaming engine starts monitoring and flowing the status of the patient data continuously.

3.4. Tier IV: Routing Data from Remote Server to Fog Controller. The PDA-doctor mapped information from the

remote server is categorized according to the status (normal, abnormal, or critical). Normal data is directly stored in the database. The fog controller has a decision time interval configured for abnormal and critical states. The abnormal and critical data from the corresponding PDA is continuously sent to the appropriate hospitals and expert doctors. For the critical data, the patient status and the location information were sent to the caregivers. Fog controller does three jobs that are the geographical distribution of patient data with less latency, localized decision mapping, and

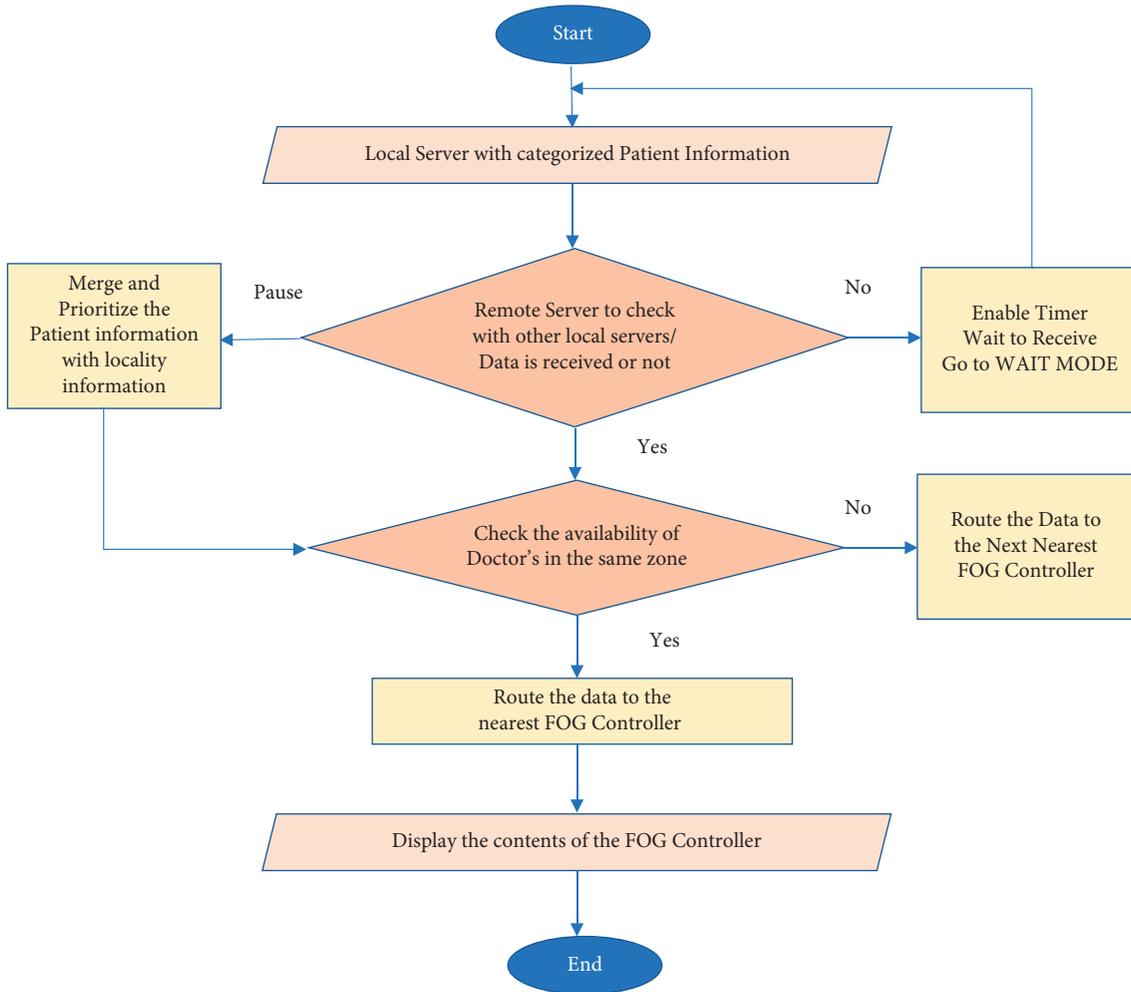


FIGURE 4: TIER III: from local server to remote server; mode of communication: wireless internet.

appropriate load balancing based on the available time of the doctor in the nearest location. If the expertise physician is engaged with some critical tasks, the next available physician in the nearest location has to be referred by the fog controller in an optimized way with reduced latency as the time required for providing the healthcare services is less compared to the cloud services as represented in Figure 5.

3.5. Tier V: Routing Data from Fog Controller to Private Cloud. The data received by the fog is routed towards the private cloud as represented in Figure 6 after the specific tasks have been completed.

- (1) Categorize patient data rendering to the locality of the server
- (2) Categorize patient data conferring to the criticality of data
- (3) Categorize patient data bestowing to the availability of doctor
- (4) Correlate the patient parameters per flag values
- (5) Notify the respective hospital or doctor based on criticality and availability.

- (6) Immediate action to both yes or no cases as per flag values so as the patient can get the medical attention without time lag

Figure 7 presents the timely remote medical diagnosis system proposed with fog essentials, fog reduces the tasks of the cloud and takes local decisions then and there and provides the required data to the needy patients regardless of the geographic boundaries globally.

An enhanced hospital management system swearing WBAN with the patient in an autonomous and also in an emergency situation is represented in Figure 7. The fog controller mechanism has been introduced in the architecture of CPS. The WBAN circuit that is worn by the patient is received by the personal data collector. All the heterogeneous data collected from different sensors in different units were digitized. The digitized data is made as a data packet with additional security code, header, and other priority relevant data that is used to emphasize the urgency of the patient to be given preference over the other continuously monitoring patients. This data sequence was collected by the local server in the same hospital. Simultaneously, the same data was sent to the remote servers in the adjacent localities based on the location of the current and

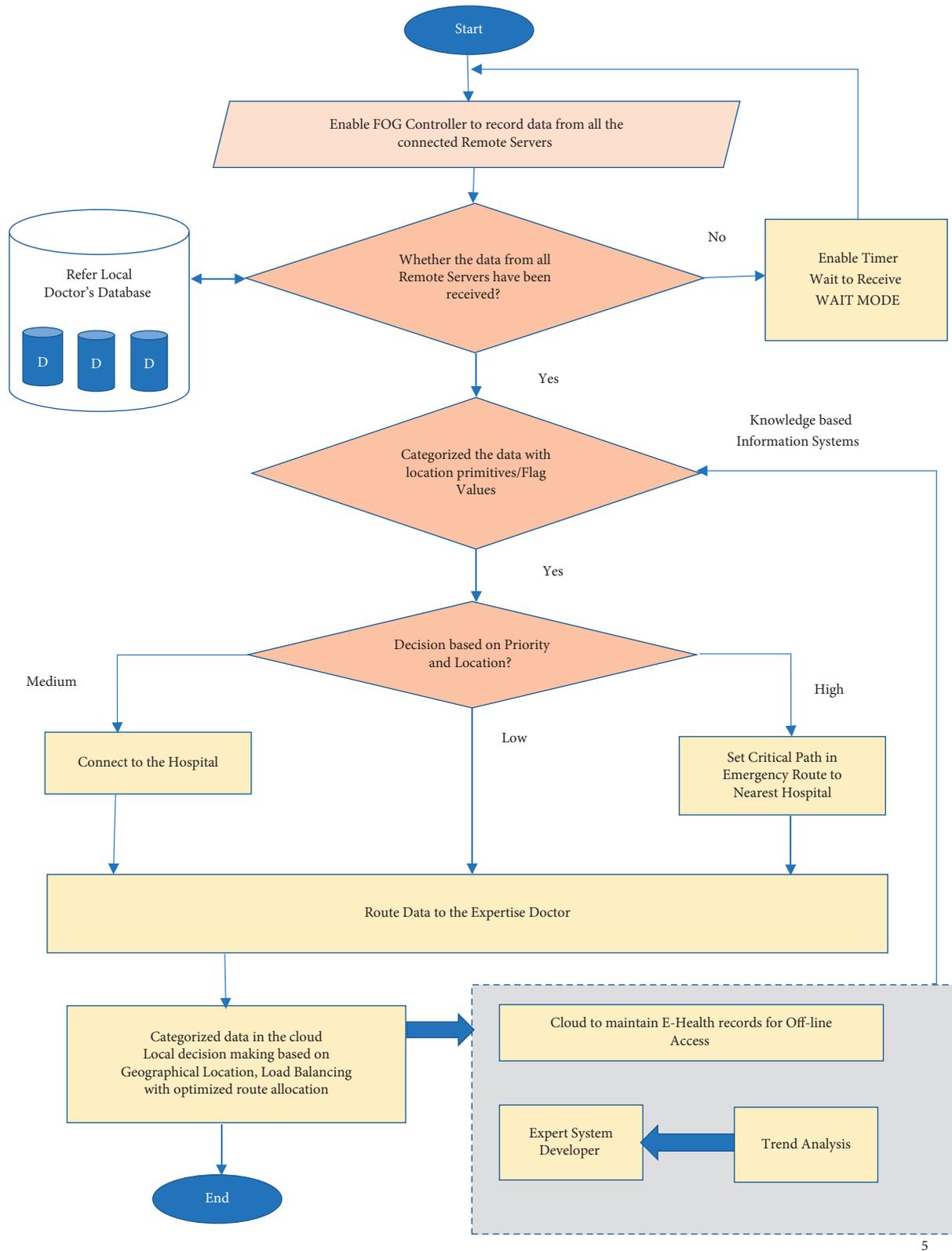


FIGURE 5: TIER IV: from remote server to fog controller; mode of communication: wireless internet.

adjacent hospitals and the readiness of doctors to attend to the patient. The remote servers were available in a sufficient number of hospitals and not in all the hospitals where all the

local servers operate. Then the data from the remote server travels via the fog controller where the decision-making is done based on geographical distances and the emergency of

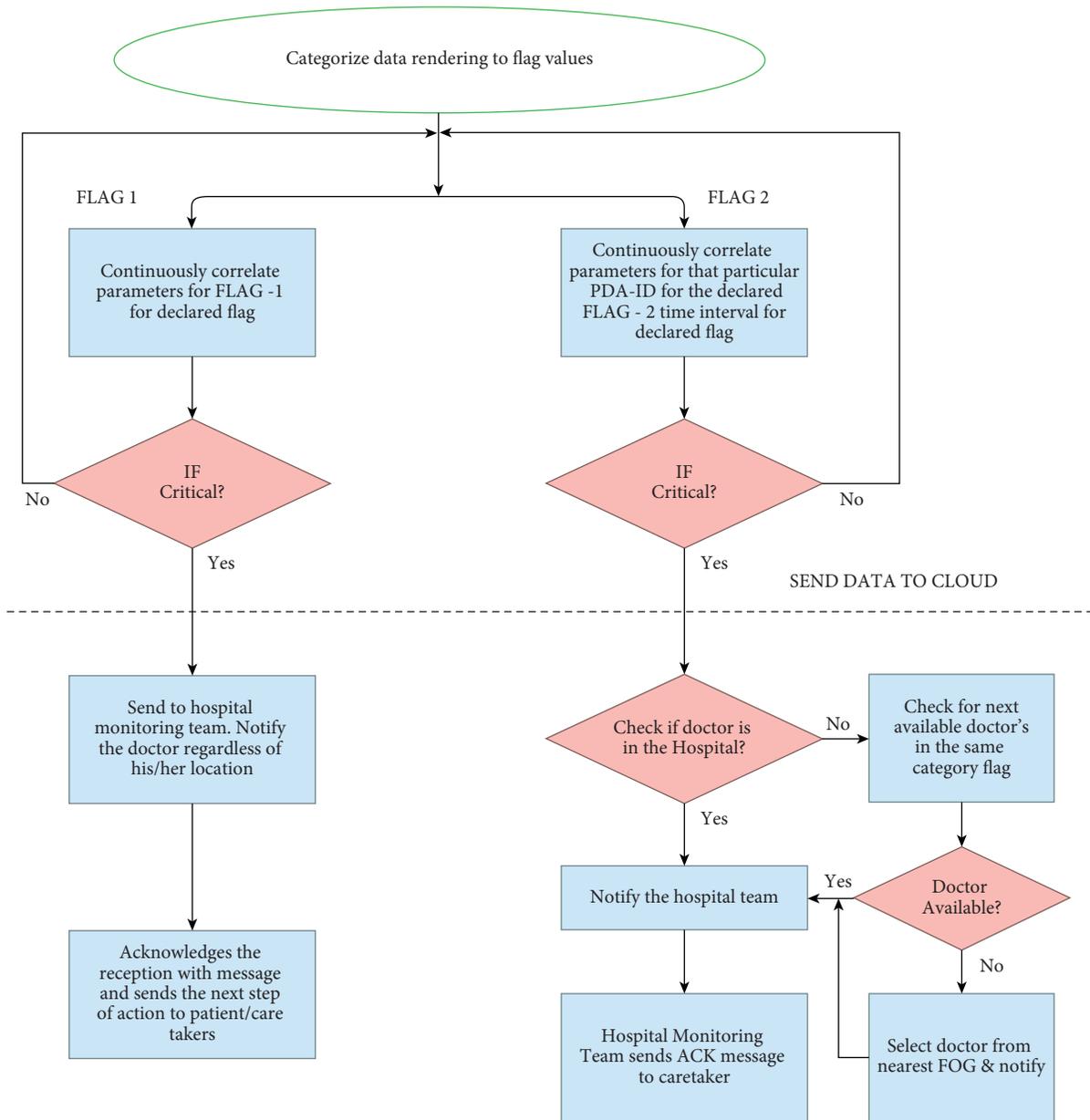


FIGURE 6: TIER V: from remote server via fog controller to cloud; mode of communication: wireless internet.

the patient data along with the availability of the physicians in the nearby hospitals. Finally, the data reaches the cloud.

The characteristics of fog have been defined as low latency and location awareness, widespread geographical distribution, network mobility with a large number of interoperable nodes, and predominant wireless access with real-time streaming capability with heterogeneous applications [30, 31]. The requirement of a smart hospital for providing accurate and timely healthcare architecture using CPS was proposed by the authors of [32, 33]. The concept of the smart city involving multiple disciplines like smart community, smart transportation, smart healthcare, and smart parking was proposed with the proposed architecture [34].

Some reliable transport layer protocols that provide end-to-end reliability of data transmitted in healthcare wireless sensor networks, and the advantages and disadvantages of

MAC, routing, and transport layer protocols have been proposed by the authors of [35, 36]. The paper reviews the existing schemes on security solutions in wireless healthcare scenarios with a summary of open security research issues that need to be explored for future healthcare applications using WMSNs [37, 38].

The authors have proposed software architecture for information sharing and collaboration that dealt with analysis, modeling, and development. It also elucidates the interoperability in sharing the medical records with the semantic details [39]. The authors proposed an IoT-based new semantic interoperability model (IoT-SIM) to provide semantic interoperability among heterogeneous IoT devices in the domain of healthcare. The doctors communicate to their patients with IoT devices to monitor the patients' current health conditions, and the information between the

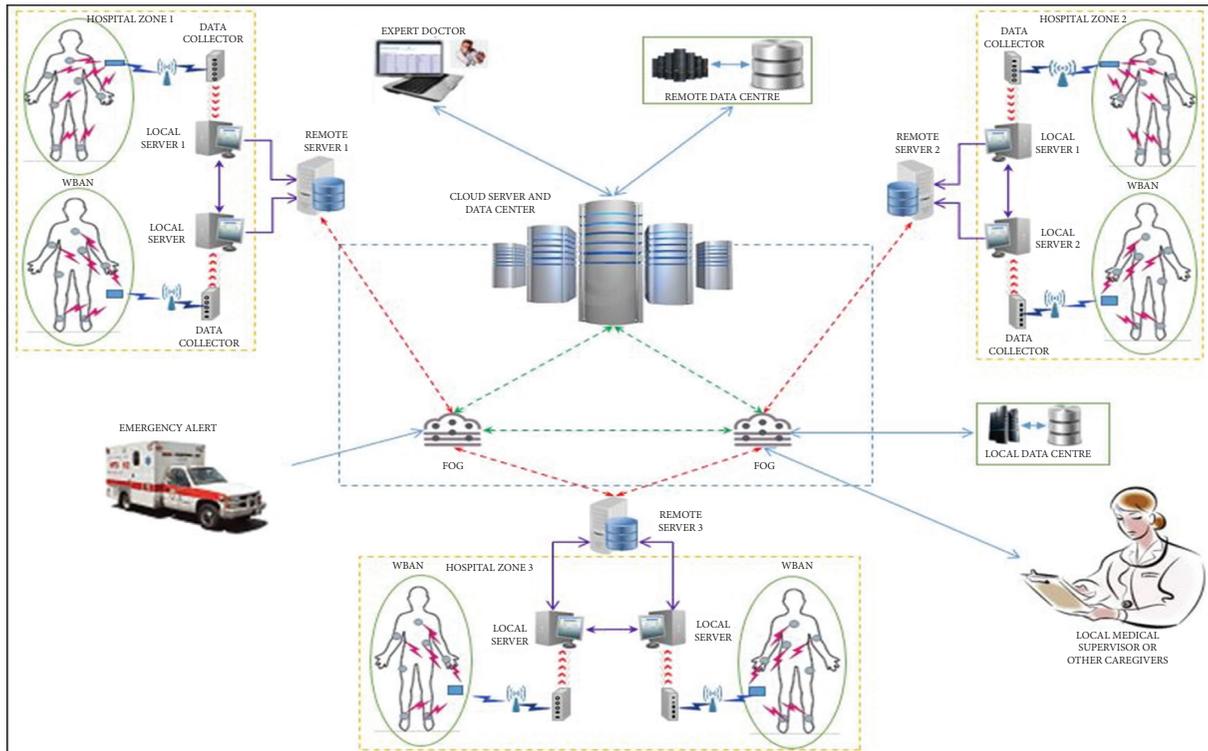


FIGURE 7: WBAN with fog essentials swarming internet of things.

doctors and the patient is semantically annotated and communicated. The semantic web technologies provide the tools that allow to process data more effectively and accurately, create the framework for interoperability between HS, and also integrate data from various sources with their semantic meaning [40]. The doctor allocation process from Tiers II–V is illustrated in Figure 8.

4. An Application Framework for Cyber-Physical System with Ganglia

To investigate system load and connectivity issues, we use the Ganglia monitoring tool. Every instance to be monitored should run the ganglia monitoring daemon (gmond). Aggregated data instances should contain additional packages. There is only one configuration file per cluster, and the configuration file does not change as long as the server is active. New instances are discovered immediately and terminated instances are forgotten within 90 seconds.

Figure 9 indicates the architecture for the system, which fuses the physical and cyberspaces. Whenever a new instance comes up, the monitoring tool will discover all instances in its cluster and send metrics to them. A newly launched instance is discovered and added to a cluster immediately. It will also do a rediscovery every 90 seconds so that instances that have been terminated are removed from its list of destinations as shown in Figure 9.

4.1. Cyber-Physical System. Cyber-physical system is a driving force for interconnecting the physical and cyber

world. It is the basis for connecting wireless sensors to the cloud even. There is plenty of scope for WSN acting as a wireless body area network. The term cyberspace is used for managing physical spaces. But it is applied to the virtual space that is created within the Internet. Cyberspace is more than a symbolic and figurative space that exists on the Internet. As cyber-physical systems are developed with the products, sensors, equipment, systems, hardware, software, and API, they can bring ubiquity everywhere in the world.

A Sensor-Cloud can be one of the pervasive applications using sensors as an interface between the physical and cyber world.

Sensor-Cloud is a part of cloud computing that uses the physical sensors for different sensing modalities to accumulate its data and transmit all the sensor data into a cloud computing infrastructure. Sensor-Cloud handles the sensor data efficiently, which is used for many monitoring applications. We use Sensor-Cloud for the WBAN application. The following points specify the necessity of a Sensor-Cloud:

- (1) To acquire the data from sensors, sensor modeling language can be used. The metadata is very important, as sensor data without location is meaningless in a sensor-centric world.
- (2) The Sensor-Cloud architecture provides instances as created from the event-driven mechanism that is devised as virtual sensors, which are going to travel in the network based on the requirement and the emergency of the patient's data.
- (3) These virtual sensors can be put into the cluster CPU, cluster memory, cluster network, cluster process,

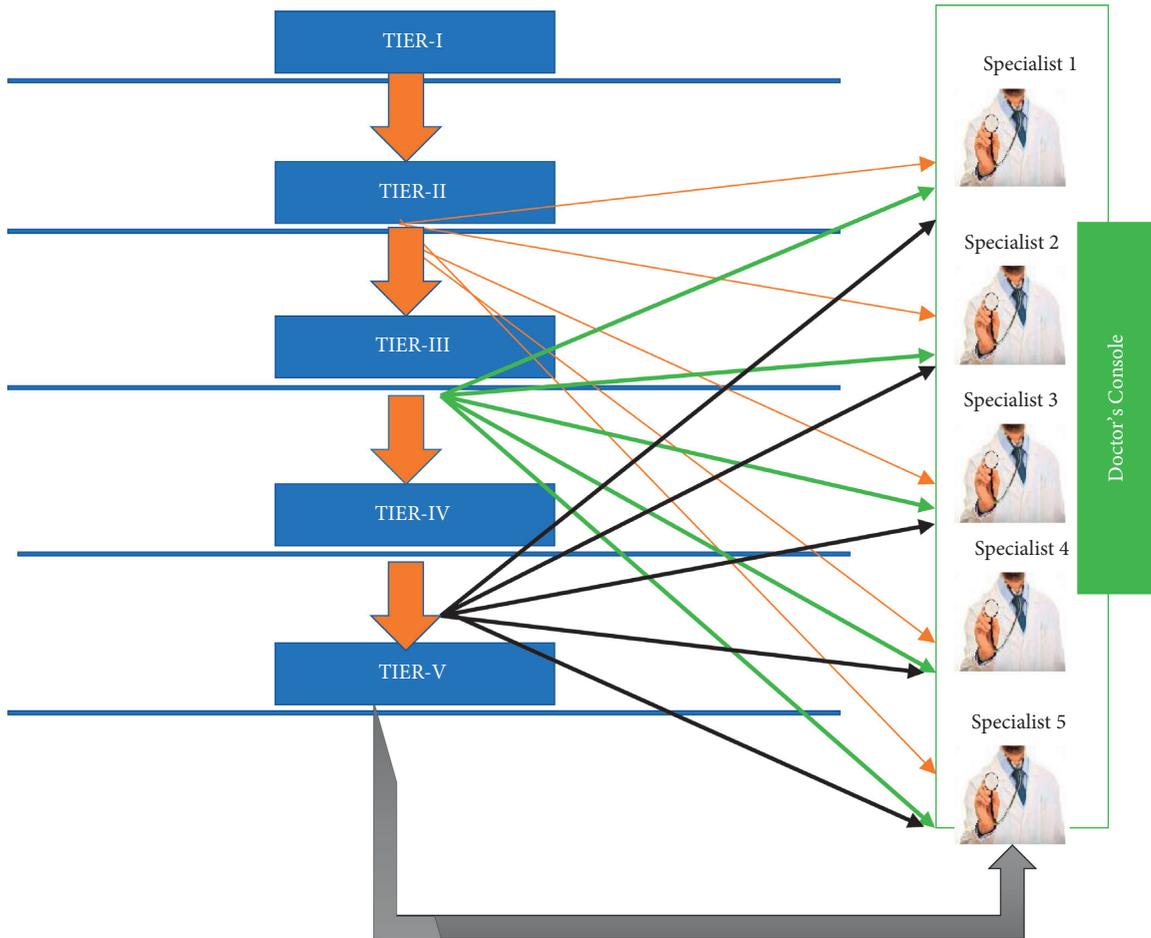


FIGURE 8: Doctor allocation process.

cluster storage, and so on. These service instances can be used through the graphical user interface.

- (4) Once the service instances become useless, they can be deleted by the users to avoid changes.
- (5) Automation of service played a crucial role in providing cloud computing sessions.
- (6) The performance metrics of the cloud can be categorized as efficiency, flexibility to avail cloud services, the delivery time of the service with tacit, and explicit knowledge.
- (7) Using the instances properly, the allocation of resources was done incorporating a policy-based resource provisioning control mechanism through software.
- (8) High importance is given to healthcare as per IOM (Institute of Medicine).
- (9) Fidelity of the system is considered as the most important criterion because of the amount of data collected from the sensor field and delivered to the intended person within the stipulated time and as it is also dependent on the application.
- (10) As fidelity is a critical factor, it means receiving the required amount of data packets to detect features of interest in a given time instant.

- (11) The intensity variations in sensor data can be used to model the trend analysis.
- (12) To route the data packets, the greedy perimeter stateful routing technique can be applied as it forwards the packets to nodes that are close to their destination. Also, it needs knowledge about the geographical coordinates of the sensor net.
- (13) In regions where such a greedy path does not exist, the node can deliver the data to the perimeter node where the packet travels successfully through the planar region of the network until it reaches a node closer to the destination.
- (14) When there is any network problem, it must be resilient to withstand and recover quickly from different complicated conditions.

4.2. Objectives of Cyber-Physical System Establishment. The following five objectives have been conceived to implement the cyber-physical system to address the scalability and reliability issues in remote medical diagnosis:

- (1) Design and development of a data acquisition system with threshold detection policies

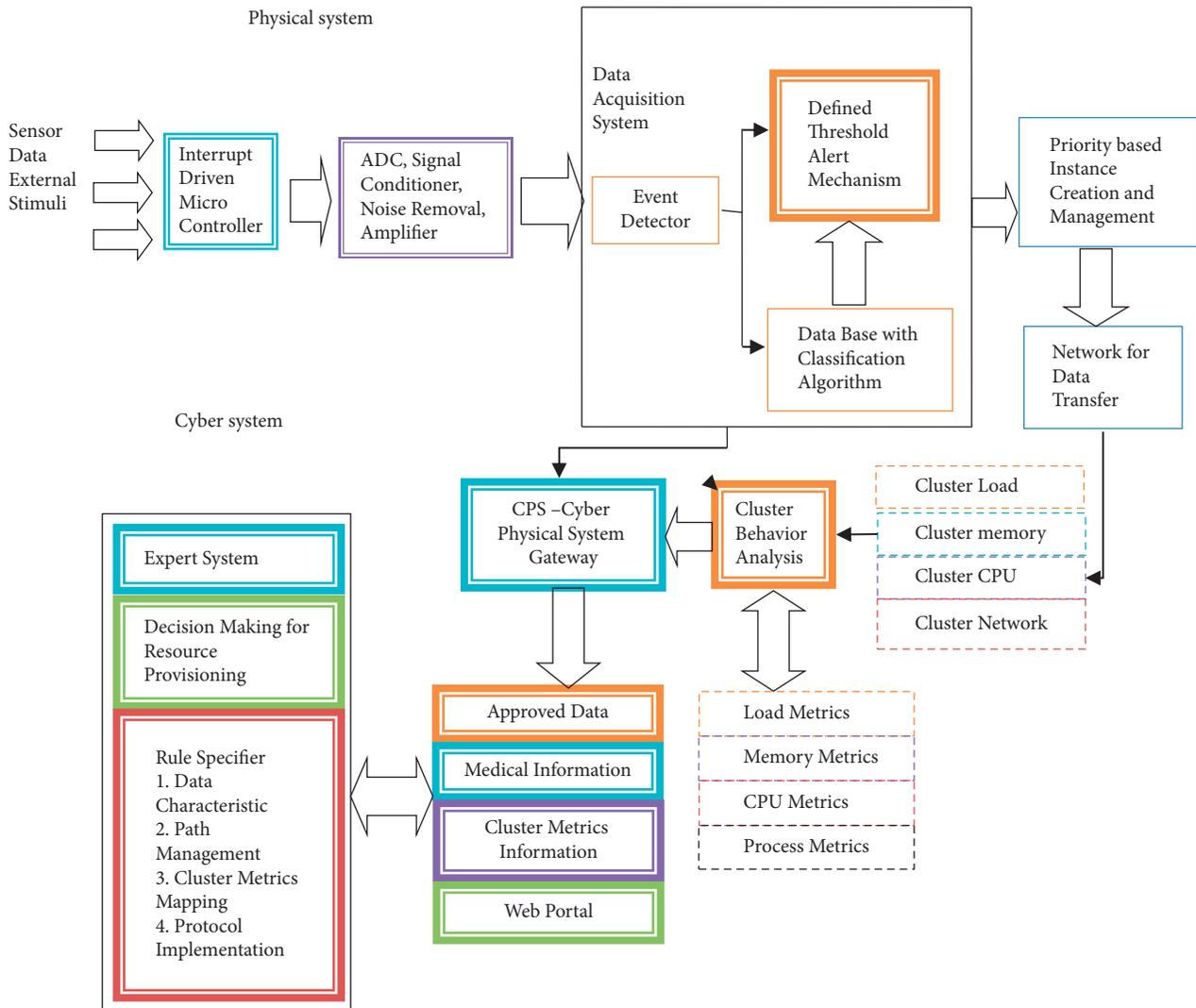


FIGURE 9: Architecture for cyber-physical system (CPS), CloudSense.

- (2) Design and development of a software-defined controller to analyze the network characteristics
- (3) Development of a path planning strategy for time-critical data
- (4) Development of a knowledge base for patient data and network data correlation aspects
- (5) Design and development of a mapping table to map the input events with network instances based on priority queuing

4.3. *Description of CPS Architecture.* The building blocks of the architecture are as follows.

4.3.1. *Physical System.* The physical system comprises the physical wireless body area sensors connected to the appropriate microcontroller units with required signal conditioning mechanisms. The noise removal and amplification aspects are carried out so that the output from one unit of the WBAN is given as one event from the physical device with

the timestamp. The sensors in the WBAN unit were of different modalities such as temperature, blood pressure, skin conductance, oxygen saturation (SPO_2), and ECG. These parameters were in different units and different formats. The data packet formation for the medical data has to incorporate all the sensor data in the digital format with source ID, destination ID, sequence no., and length payload with security codes. The sample format is shown in Figure 10.

4.3.2. *DATA Acquisition Systems (DASs).* The DAS comprises an event detector, defined threshold alert mechanism, and a database with a classification algorithm built-in. An event detector is a mechanism through which if any input signal has to be attended to and when the physiological body parameters have been accessed. Defined threshold alert mechanism consists of a two-way threshold-based alerting where the input signal that goes beyond the critical level was identified. Identification of the threshold value is based on the comparison between the normal body parameter values

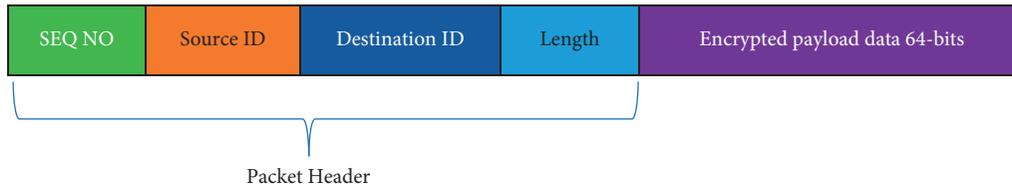


FIGURE 10: Data packet format.

and the measured body parameter values. The threshold monitoring tool has to sit all day long watching and waiting for the event to occur based on the comparison. This is a monitoring system that will alert when the body parameter values become critical. If the disease is identified properly, proper physicians were connected without delay.

(1) *Database with Built-in Classification Algorithm: (Stochastic Modeling)*. Stochastic modeling is concerned with the application of probability to real-world situations characterized by uncertainty. Due to the pervasive nature of uncertainty, the tools have the potential to demonstrate their measurement quality in almost every aspect of the medical diagnosis system.

4.3.3. Priority-Based Instance Creation and Management.

The interlinking of the physical and cyber systems occurs at the priority-based instance creation and management system. This is again a mechanism that provides priority queuing based on the criticality of instances that occur. During the situation, when a critical issue occurs, based on the emergency of the human's health condition, the situations call for a queuing scheme that allows having priority over all other conditions, priority queuing (PQ) is being considered for the scenario. PQ has been allocated four queues in our study, each with a different priority: high priority, medium priority, regular priority, and low priority. After the highest priority queue has been emptied that must be served first, the data packets from each queue are transmitted. Within each queue, packets are transmitted first in, first out based on a crucial event in the calculation of human body parameters. The queue size does not affect the amount of time the packets obtain in that queue. PQ queue size is optimized for data packets. In order of priority, each queue is served. The high-priority queue is often first served because it concentrates on emergency data packets; if the high-priority queue is empty, then the medium queue is emptied and also serves next to the patient's critical condition.

Whenever a high queue packet, which is an emergency data packet, is received, the doctor must attend immediately. The queue is filled before any other queues are processed. When the medium-priority queue is emptied, the usual queue that holds normal body parameter values is emptied if there are no packets in the high-priority queue. Finally, the low-priority queue is emptied if the high, medium, and regular queues are emptied. Therefore, when PQ is used, packets in lower priority queues cannot be forwarded in the required time, causing network apps to run out of time for

applications with packets using lower priority protocols. If a packet does not fit any of the configured queues, the packet goes to the usual queue. Since PQ is not dynamic, it does not respond to network trends. When used, it is a good idea to conduct network baselines regularly and to review traffic to ensure that queue size and protocol distributions are correctly configured to manage the traffic at peak times shown in Figure 11.

(1) *High-Priority Queue*. Packets arriving at the high-priority queue shall immediately be served. The medium-, regular-, and low-priority queues are serviced after the high-priority queue has been emptied. If packets arrive for the high-priority queue at any time, they were transmitted before the high-priority queue has been emptied before any other queue receives operation. The high-priority queue's default size is 20 packets.

(2) *Medium-Priority Queue*. The medium queue is serviced after the high-priority queue has been emptied. When a high-priority queue arrives, the packets in the high-priority queue are forwarded first, until the queue is empty, and then the medium queue receives attention again. The medium-priority queue's default size is 40 packets.

(3) *Normal Priority Queue*. If the high or medium queues have no packets, the usual queue is serviced. When packets arrive at the high or medium queues, these are forwarded to medium in order, and packets are sent to the usual queue after those queues have been cleared. The normal priority queue size is 60 packets. By default, all unspecified traffic is allocated to the usual priority queue, but by using the default argument, you can change this behavior.

(4) *Low-Priority Queue*. Low-priority queue packets were forwarded if all other queues are empty. When a packet comes in any of the other queues, the queues were cleared first, until they are empty, and then the low-priority queue is serviced back. The low-priority queue default size is 80 packets.

A priority list is created to configure PQ. To create a priority list, 16 priority lists can be created. There are 4 queues in each list: high, medium, regular, and low. Packets are allocated to one of the four queues depending on their features: the protocol, the input interface, the size of the packet, the criticality factor, and patient position. Traffic not specified in one of the four queues is sent to the standard queue, the normal queue. The priority-list command, its arguments, keywords, and descriptions are included in Table 1.

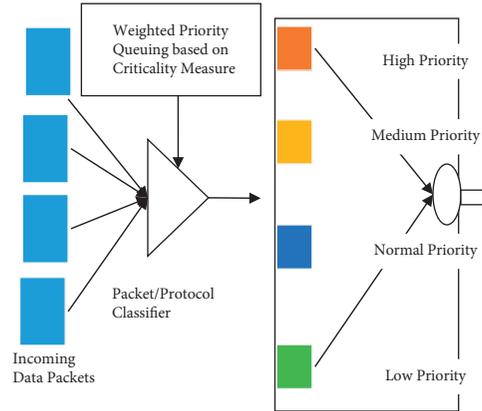


FIGURE 11: Priority-based weighted queuing mechanism for data packet transfer.

TABLE 1: Sample priority table.

Priority list command (listing number)	Arguments (parameters)	Keywords (percentage of severity) (%)	Criticality factor (critical value)	Timestamp (time of occurrence)	Location ID of the patient (country, state, and IP address)	Assignment of queue (name of queue)
1001	ECG, BP	100	0	9:00 a.m.	Inchlocid01	High
1002	BP, TMP	75	1	12:00 p.m.	Pkpelocid45	Medium
1003	SPO ₂ , TMP	50	2	2:45 a.m.	Uscalocid56	Normal
1004	SC, TMP	25	3	4:54 p.m.	Rumalocid98	Low

A sample priority table consists of the following information:

- (1) Priority listing is the event number, based on the incoming traffic received from ingress
- (2) Arguments consist of parameter listing, based on the combination of parameters for the cause of disease and the attendance requirement of the physician
- (3) The keywords column indicates the percentage of the severity of the disease and the need for attention
- (4) Criticality factor presents any one of the four values 0, 1, 2, and 3 depending upon the queue in which the data has to fit in for a diagnosis
- (5) The timestamp indicates the incoming data time, based on the time of occurrence of the event, as it is possible to predict the time from the wireless body sensor data unit (from the data collector within the body sensor unit)
- (6) Location ID consists of the country code, state code, and the IP address of the system from which the patient's information is received
- (7) Assignment of queue indicates the name of the queue through which the patient's information was processed for diagnosis

Table 1 presents the fields such as priority-list command like the patient ID, the parameters that are observed from the patient, the severity of the disease from the observed parameter, the critical value that indicates whether the patient has to be admitted immediately by the doctor, timestamp of the data, location data that includes the country name, state name, and the IP address of the system from where the data

is being routed, and the importance of the queue from where the data have to be fetched.

To assign a public IP address to the EC2 instance, the following procedure is adopted:

- (1) Open the Amazon EC2 console.
- (2) In the navigation pane, choose the Instances icon.
- (3) Select the instance of your own, and choose the Actions icon, Networking, and Manage IP addresses.
- (4) Expand the network interface. Under IPv6 addresses, choose Assign new IP address.
- (5) Then choose Save.
- (6) To assign an ephemeral external IP address that does not persist beyond the life of the resource, we create an instance or forwarding rule without specifying an IP address, the resource is automatically assigned an ephemeral external IP address.
- (7) Spot instances can also be created.

5. Mathematical Modeling

The list of physiological parameters that can be observed through the body sensor network is categorized as follows:

$$\sum_{i=1}^n \text{Phy.Pi} \quad (1)$$

Phy.Pi consists of the five vital body parameter values considered for this research. ECG is modeled as A ; body temperature is modeled as B ; and oxygen saturation is modeled as C .

- (1) ECG can be measured from the human in the range of $+o$ to $-o$, where all the variations observed from the human's ECG values comprise the two extremities of death conditions and all the living conditions before death in both extremes. When we normalize the range of ECG, a human can withstand; it has the maximum ECG value when death can occur as well as the minimum ECG value at where death can occur. The maximum ECG value is normalized as $+1$, and the minimum ECG value is normalized as -1 . The normal ECG value, a human can withstand, is termed as 0 . The acronym ALD indicates all ranges of ECG values including the two extremities of death conditions along with the normal life condition and the intermediate life conditions between normal and $+1$ and between normal and -1 . These between life conditions have been viewed as disease conditions and ready for diagnosis.

$$A_{LD} = \int_{-o}^{+o} d(\text{NECG}). \quad (2)$$

Equation (2) indicates the total number of ECG values that can be measured from a human at any time instant.

- (2) Body temperature can be measured from the human in the range of $+p$ to $-p$, where all the possible temperature values that can be observed from the human can include both the extreme values of temperature meeting death in maximum and minimum level, which are $+1$ and -1 . Along with the extreme values, the normal human body temperature's normalized value is 0 . Between 0 and $+1$ and between 0 and -1 , both intermediate values will lead to diseases. These values need immediate diagnosis based on their level of severity.

$$B_{LD} = \int_{-p}^{+p} d(\text{NTMP}). \quad (3)$$

Equation (3) indicates the total number of temperature values that can be measured from a human at any time instant.

- (3) Oxygen content- (SPO_2) can be measured from the human in the range of $+q$ to $-q$, where all the possible SPO_2 values that can be observed from the human can include both the extreme values of oxygen saturation meeting death in maximum and minimum level, which are $+1$ and -1 . Along with the extreme values, the normal human's oxygen saturation's normalized value is 0 . Between 0 and $+1$ and between 0 and -1 , both intermediate values will lead to diseases. These values need immediate diagnosis based on their level of severity.

$$C_{LD} = \int_{-q}^{+q} d(\text{NSPO2}). \quad (4)$$

Equation (4) indicates the total number of Oxygen saturation values that can be measured from a human at any time instant.

$$D_{LD} = \int_{-r}^{+r} d(\text{NBP}). \quad (5)$$

- (4) Blood pressure can be systolic and diastolic and be measured from the human that indicates the pumping mechanism of the heart.

Equation (5) indicates the total number of blood pressure values containing a separate column of low blood pressure and high blood pressure. The extreme values for the human's death conditions about maximum and minimum values along with the in-between values from the normal value in both directions have to be observed as 0 to $+1$ and 0 to 1 .

$$E_{LD} = \int_{-s}^{+s} d(\text{NSC}). \quad (6)$$

Pulse oximetry measures the amount of oxygen being carried in our blood, as a percentage that can be measured at the finger using a pulse oximeter. This measurement is known as the SPO_2 and the saturation of peripheral oxygen is SAO_2 , which is the saturation of arterial oxygen. A decrease in oxygen saturation and increases in pulse rate and heart rate variability have been found to be associated.

- (5) Skin conductance: equation (6) presents the total number of skin conductance values about the normal skin conductance value as zero and the extreme values in both directions indicating death as $+1$ and -1 . Excluding the normal value, the entire set of values has to be diagnosed with prioritization parameters.

Excluding the death conditions from equations (2)–(6), the following can be the equations for life and diagnosis:

$$A_L = \int_{-o}^{+o} d(\text{NECG}) - [\text{NECG}_{+o}, \text{NECG}_{-o}], \quad (7)$$

$$B_L = \int_{-p}^{+p} d(\text{NTMP}) - [\text{NTMP}_{+p}, \text{NTMP}_{-p}], \quad (8)$$

$$C_L = \int_{-q}^{+q} d(\text{NSPO2}) - [\text{NSPO2}_{+q}, \text{NSPO2}_{-q}], \quad (9)$$

$$D_L = \int_{-r}^{+r} d(\text{NBP}) - [\text{NBP}_{+r}, \text{NBP}_{-r}], \quad (10)$$

$$E_L = \int_{-s}^{+s} d(\text{NSC}) - [\text{NSC}_{+s}, \text{NSC}_{-s}]. \quad (11)$$

5.1. Case I: Instance Creation for Parameter 1 – ECG. As mentioned, $\text{ins}_{t_0} - \text{ins}_{t_t}$ indicates the life and death instances about ECG. Instances between $+o$ and $-o$ can be termed as t .

All the ECG instances that we can expect become, ins_{ot_0} , ins_{ot_1} , ins_{ot_2} , \dots , ins_{ot_t} from the time instant t_0 to t_t .

Time instants of occurrences are represented in

$$ECG = \int_{ins_{ot0}}^{ins_{ott}} d(A_{LD}), \quad (12)$$

where A_{LD} is represented in equation (2).

A random sample of any of the ECG instances is exhibited in the equation. Equation 12 shows a random sampling of any of the ECG instances. Any subsample of ECG instances that requires critical diagnosis is created from the random sample.

Immediate diagnosis of ECG is represented in

$$ECG_{IM} = \int_{ins_{ot0}}^{ins_{ott}} d\{A_{LD}\}_{CF}. \quad (13)$$

5.2. Case II: Instance Creation for Parameter 2 – Temperature. As mentioned, $ins_{to}-ins_{tt}$ indicates the life and death instances about human body temperature. Instances between $+p$ and $-p$ can be termed as t .

From the time instant t_0 to t_t , all the temperature instances that we can predict become $ins_{pto}, ins_{pt}, ins_{pt2}, \dots, ins_{tt}$. Temperature instances are represented in

$$TMP = \int_{ins_{pt0}}^{ins_{ptt}} d(B_{LD}), \quad (14)$$

where B_{LD} is represented in equation (3).

A random sample of any of the temperature instances is exhibited in the equation. Equation 14 shows a random sampling of any of the temperature instances. Any subsample of temperature instances that requires critical diagnosis is created from the random sample. Immediate diagnosis of TMP is represented in

$$TMP_{IM} = \int_{ins_{pt0}}^{ins_{ptt}} d\{B_{LD}\}_{CF}. \quad (15)$$

5.3. Case III: Instance Creation for Parameter 3 – SPO₂. As mentioned, $ins_{to}-ins_{tt}$ indicates the life and death instances about human body oxygen saturation. Instances between $+q$ and $-q$ can be termed as t .

From the time instant t_0 to t_t , all the SPO2 instances that we can predict become $ins_{qto}, ins_{qt1}, ins_{qt2}, \dots, ins_{qtt}$.

The oxygen saturation instances are represented in

$$SPO2 = \int_{ins_{qt0}}^{ins_{qtt}} d(C_{LD}), \quad (16)$$

where C_{LD} is represented in equation (4).

A random sample of any of the oxygen saturation instances is exhibited in the equation. Equation 16 shows a random sampling of any of the SPO2 instances. Any subsample of SPO2 instances that requires critical diagnosis is created from the random sample. Immediate diagnosis of SPO₂ is represented in

$$SPO2_{IM} = \int_{ins_{qt0}}^{ins_{qtt}} d\{C_{LD}\}_{CF}. \quad (17)$$

5.4. Case IV: Instance Creation for Parameter 4 – Blood Pressure. As mentioned, $ins_{to}-ins_{tt}$ indicates the life and

death instances about low or high blood pressure values. Instances between $+r$ and $-r$ can be termed as t .

From the time instant t_0 to t_t , all the blood pressure instances that we can predict become $ins_{rto}, ins_{rt1}, ins_{rt2}, \dots, ins_{rtt}$.

The blood pressure instances are represented in

$$BP = \int_{ins_{rto}}^{ins_{rtt}} d(D_{LD}), \quad (18)$$

where D_{LD} is represented in equation (5).

A random sample of any of the blood pressure instances is exhibited in the equation. Equation 18 shows a random sampling of any of the blood pressure instances. Any subsample of blood pressure instances that requires critical diagnosis is created from the random sample. Immediate diagnosis of blood pressure is represented in

$$BP_{IM} = \int_{ins_{rto}}^{ins_{rtt}} d\{D_{LD}\}_{CF}. \quad (19)$$

5.5. Case V: Instance Creation for Parameter 5 – Skin Conductance. As mentioned, $ins_{to}-ins_{tt}$ indicates the life and death instances about low or high skin conductance values. Instances between $+s$ and $-s$ can be termed as t . From the time instant t_0 to t_t , all the skin conductance instances that we can predict become $ins_{st0}, ins_{st1}, ins_{st2}, \dots, ins_{stt}$.

The skin conductance instances are represented in

$$SC = \int_{ins_{st0}}^{ins_{stt}} d(E_{LD}), \quad (20)$$

where E_{LD} is represented in equation (6).

A random sample of any of the skin conductance instances is exhibited in the equation. Equation 20 shows a random sampling of any of the skin conductance instances. Any subsample of skin conductance instances that requires critical diagnosis is created from the random sample. Immediate diagnosis of skin conductance is represented in

$$SC_{IM} = \int_{ins_{st0}}^{ins_{stt}} d\{E_{LD}\}_{CF}. \quad (21)$$

Combining all the critical instances for the mentioned five parameters become

$$\begin{aligned} Z = & \int_{ins_{ot0}}^{ins_{ott}} d\{A_{LD}\}_{CF} + \int_{ins_{pt0}}^{ins_{ptt}} d\{B_{LD}\}_{CF} \\ & + \int_{ins_{qt0}}^{ins_{qtt}} d\{C_{LD}\}_{CF} + \int_{ins_{rt0}}^{ins_{rtt}} d\{D_{LD}\}_{CF} \\ & + \int_{ins_{st0}}^{ins_{stt}} d\{E_{LD}\}_{CF} + . \end{aligned} \quad (22)$$

The questions arise like, how to include the entire set of critical instances that have to be given high prioritization. How to handle if the number of virtual instances has come more in terms of CPU, memory, network, cluster usage, and load/process time, which is shown in Figure 12.

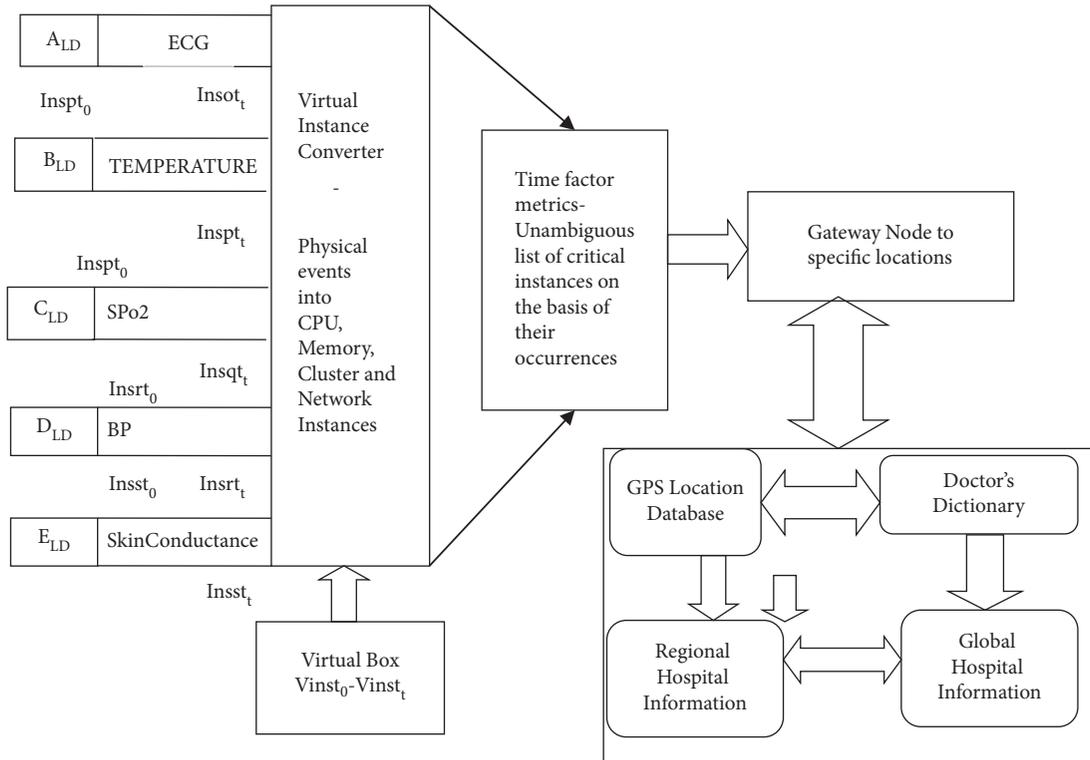


FIGURE 12: Instance creation and management for smart health data.

6. Addressing the WBAN: Data Scalability Issue

6.1. *Cyber System Components.* Eucalyptus enables pooling compute, storage, and network resources that can be dynamically scaled up or down as application workloads change. It is compatible with Amazon’s EC2.

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure, resizable compute capacity in the cloud. An Elastic Compute Cloud instance is a virtual server that can run applications in Amazon Web Services (AWS). When setting up an EC2 instance, we can custom configure CPU, storage, memory, and networking resources, and when an instance is created, we can create it with an Amazon Machine Image (AMI).

Ganglia gets installed on the cluster at the bootstrap stage that provides insight into how each box in the cluster is performing. If the CPU was running high, it might be wise to choose EC2 instances that had larger cores, or if the memory was overutilized also, then EC2 instances can be chosen with a larger memory capacity.

6.2. *Image Management in CloudSense.* At this juncture, the cloud is ready to operate. The node at “192.168.20.2” was registered, and the resource the cloud has to offer is within the availability zone of the cloud. The type m1 small indicates that there are four machines available and can be created each with one processing unit, RAM 192 MB, and a disk with 2 GB space. The VM images can be downloaded from the Eucalyptus store. Eucalyptus has provided links to

prepackaged virtual machines that are ready to run in the private Eucalyptus cloud.

6.3. *Instance Management.* The images have been registered. They are now ready to be uploaded and operated upon. An instance can be sprung up after the assignment of a public IP by the DHCP daemon. Instances can be terminated as and when they are not required.

6.4. *Cloud Monitoring.* There are myriad ways to view the performance of the cloud. Many tools have been developed. Of them, the tool “Ganglia” is a very efficient and effective tool that can be used to view the performance through various metrics.

Ganglia is a scalable distributed monitoring system [41]. It scales well with very large numbers of servers and is useful for viewing performance metrics in real time. The applications of ganglia are represented in Figure 13.

On the back end, Ganglia is made up of the subsequent components:

- (1) Gmond (Ganglia monitoring daemon): it is a small service that collects information about a node. This is installed on every server that is to be monitored.
- (2) Gmetad (Ganglia meta daemon): it is a daemon on the master node that collects data from all the Gmond daemons (and also from the other Gmetad daemons, on condition).

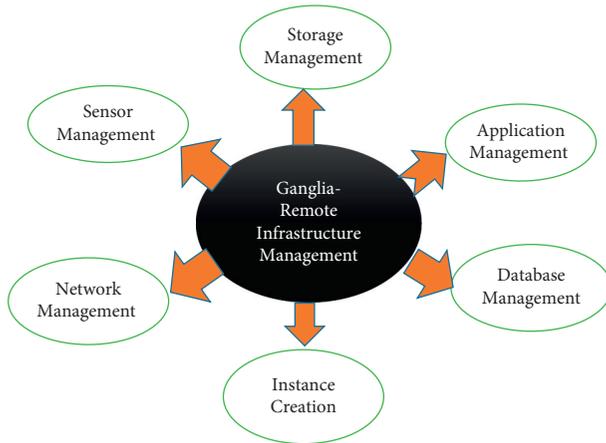


FIGURE 13: Ganglia applications.

- (3) RRD (round-robin database) tool: it is a tool on the master node used to store data and visualizations can be possible from Ganglia in the time series mode of operation.
- (4) PHP web front end: it is a web interface on the master node that displays graphs and metrics from data in the RRD tool.

Every node (server) that we want to monitor has Gmond to be installed. Every node uses Gmond to send data to the single master node running Gmetad, which should collect all the node data and send it to the RRD tool to be stored. The data in the web browser can be viewed with the help of PHP scripts and Apache.

Ganglia grid with the master node is shown as the Ganglia server running the Gmetad daemon, and the other nodes have been shown as connecting servers running the Gmond daemon:

To monitor the data through the web interface, the data is organized on several levels as shown in Figure 14. Ganglia organizes the nodes, which are individually monitored machines, into clusters, nothing but groups of similar nodes. On a higher level, collections of clusters can also be organized into grids.

Important limitations for wider acceptance of the existing WBAN systems for continuous monitoring are as follows:

- (a) Unwieldy wires between sensors and a processing unit
- (b) Lack of system integration of individual sensors
- (c) Interference on a wireless communication channel shared by multiple devices
- (d) Nonexistent support for massive data collection and knowledge discovery

Depending on the application requirements, the WBAN coordinator is further connected to telemedicine and medical servers for relevant recommendations, like connecting to the edge and fog terminals as everything can be visualized by the Ganglia monitoring system as it supports scalability in an efficient way.

Ganglia is a free and open-source monitoring tool for high-performance computing systems such as clusters and grids. It is made up of three components: gmond, gmetad, and a web front end. Gmond is a multithreaded daemon that runs on each node and communicates with gmetad. Gmetad collects and archives metrics and generates reports. Internally, it makes use of the rrdtool for data logging and graphing. The web front end included with Ganglia is used to visualize the data. Ganglia is a comprehensive monitoring solution that visualizes and archives system metrics. It comes preconfigured to monitor over 30 metrics without requiring any additional coding. The installation procedure consists of downloading ganglia packages and their dependency packages [37]. Figure 15 was taken from the web interface provided by ganglia to monitor the eucalyptus cloud.

It can be clearly understood about the moment that no heavy processes were going on, and hence, the utilization rates are quite low. The following figure indicates the cluster load as a graph, which is measured as load/process versus time. The number of node participation is 1 as minimum and 2 as maximum. The number of CPUs is 2 as minimum and 6 as maximum. The process minimum is termed as zero, and the maximum process is termed as 3. As it is not specified, it is the cluster load at the hour when it was measured in Figure 16.

As per the diagrammatic representation of cluster load last hour in Figure 16, the node considers the instance as it indicates the patient data loading. One patient, one instance is formed when the number of nodes participating is 1, and two patient nodes, two instances are created when the number of nodes participating is 2.

Figure 17 is a graph specifying the cluster memory at a given time in analyzing the cluster memory in terms of use, share, cache, buffer, swap, and total. Memory utilization can be done using the free command in Linux. The command displays the total amount of free, used physical, and swap memory in the system, as well as the buffers used by the kernel. We can also estimate the total physical memory; the total used memory, the amount of free physical memory, the size of the file cache, the total size of the swap space, the amount of swap space used, and the amount of swap space free.

The representation in Figure 18 implies that the cluster memory storage corresponds to the hospital information of patient data in the memory pertaining to the collective patient information at a particular time.

Figure 18 shows the cluster CPU usage at some time during the operation of the cluster. To monitor per-process CPU utilization, the parameters to be monitored are the CPU, memory, page faults, stack, disk I/O, context switching on a per-process for a specified time interval. The addition of the parent and child process can also be monitored using special CPU and network monitoring devices. The parameters can be indicated as threads, processes, users, and groups that are resource hogs. The amount of time that the system spent in user mode is commonly measured in units of user-HZ. User mode with low priority is observed as system mode, the idle task, and the waiting time for an I/O to finish are measured.

Ganglia Monitoring System

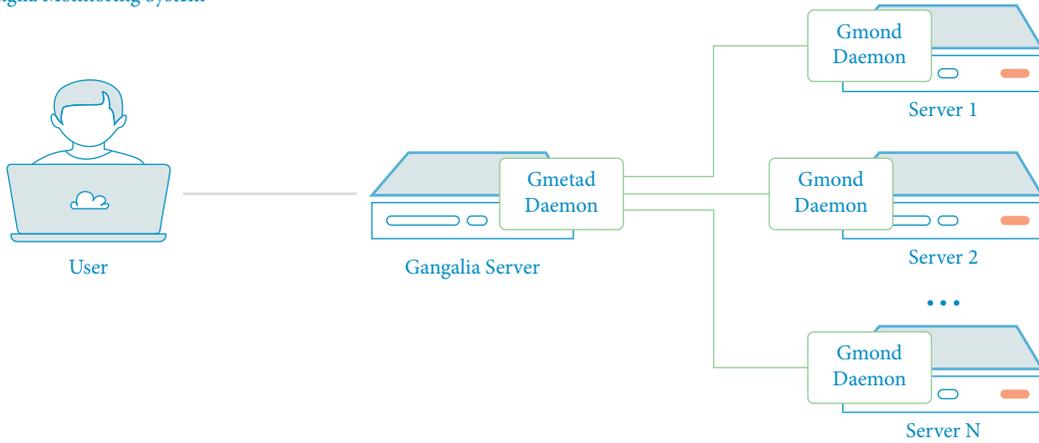


FIGURE 14: Ganglia monitoring system.

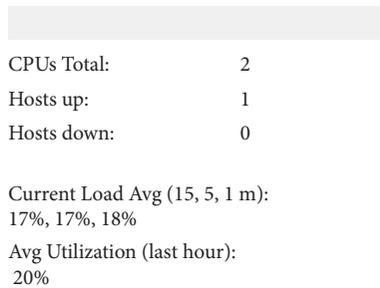


FIGURE 15: Cluster information.

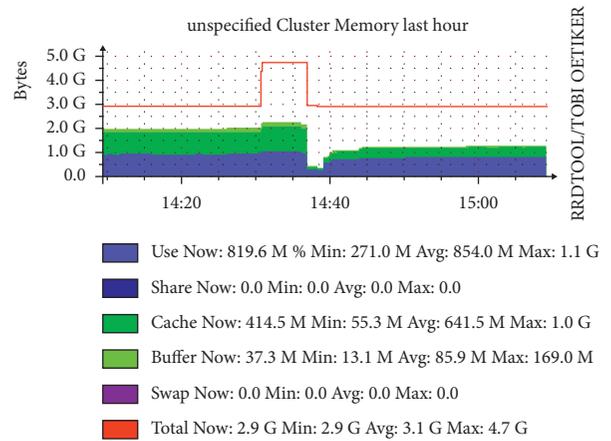


FIGURE 17: Cluster memory at a particular time.

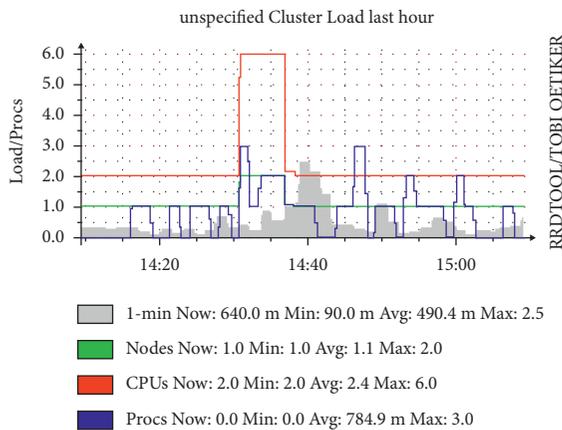


FIGURE 16: Cluster load at a particular time.

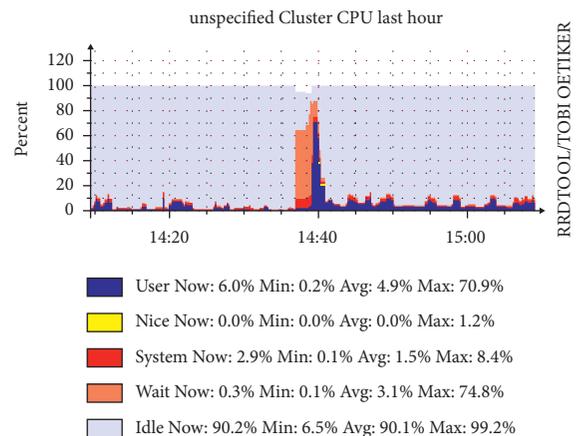


FIGURE 18: Cluster CPU usage at a particular time.

The prioritization of data pertaining to the patient’s abnormality condition is shown in Figure 18. Context switching indicates that the data can be forwarded to the port immediately as per the severity condition of the patient. If the same type of prioritized information is received, it could be mapped with parent and child processes as they fall in the same category, due to medical emergency.

Figure 19 shows the network usage over a particular hour during the operation of the cloud. The bandwidth usage by connection can be viewed as whether it is incoming traffic or outgoing traffic and which application is utilizing how much

amount of bandwidth. It can also be viewed as how many hosts are connected at a particular instant of time and what type of traffic is going on in the network.

The indication in Figure 19 corresponds to how many patients are connected to the fog at a particular time and

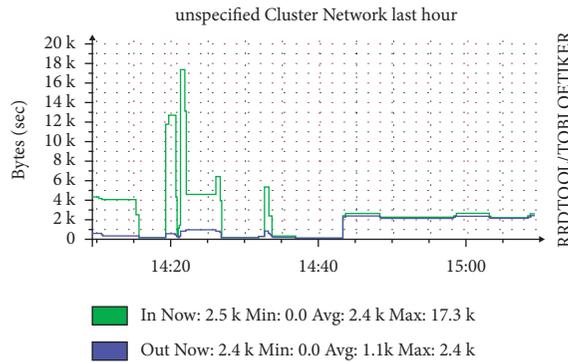


FIGURE 19: Cluster network at a particular time.

whether the patient traffic is within the bandwidth limitation or not, by comparing the incoming and outgoing traffic.

Figure 20 is the system load as it varies over time in the cluster. Clustering makes one instance of an application server into a master controller. This master controller will process all the requests and distribute them in too many numbers of instances. The algorithms used for this purpose are the normal operating system's scheduling algorithms such as round-robin and so on. Clustering is done for load balancing to enable scalability as the purpose of this research is to increase scalability when the number of inputs becomes enormous, which is shown in Figure 20. As scalability is the ability to add more instances of an application server, it increases the capacity of the system through the reduction of the response time of the application.

Figure 20 corresponds to one instance as it relates to one host like so many instances and group those instances of the healthcare application and process the data to do the data analysis and rank the data. It is highly essential to rank, in order to give importance to high-prioritized data rather than normal data, which does not require critical care.

6.5. WBAN Data. WBAN is used for people who require emergency and continuous medical care. They have made the lives of mankind a lot easier. At present, the data produced by the LS-WBAN is processed on local systems. The data is brought to the local machine via gateways and then stored on relational database systems. But the main problem is the issue of scalability. A network may increase its size over a huge geographical area. Such spatial data is massive and huge data sets are generated, which need to be crunched in real time for real-time analysis.

6.6. The Solution. Data can be streamed to the eucalyptus cloud and instantaneously stored using blocks or buckets. These data can be analyzed, and the analysis has been done by springing up instances assuming the data scales up the storage blocks in the cluster. The analysis has been done by bringing up the cluster to its limit and making a study of how the cluster behaves accordingly.

6.7. Scaling the Cluster. The cluster resources are based on the front end for storage and the creation and functioning of

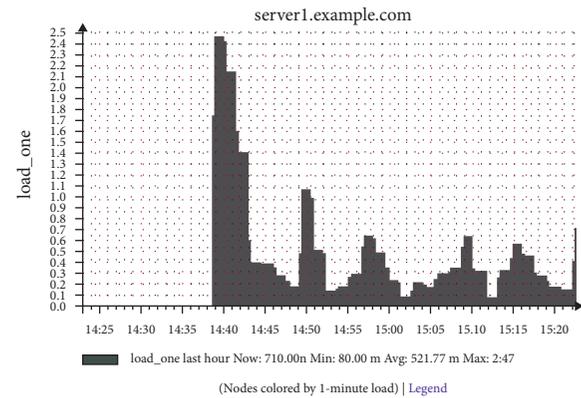


FIGURE 20: Cluster load versus time.

the VMs in the node. The nodes are brought up one by one, and the cluster behavior is studied based on the metrics provided by the ganglia interface. An instance is started using the instance command. The resources are now reduced as shown in Figure 21.

Figure 21(a): resources after starting one instance. The figure shows the load/process metric. One patient is monitored.

Figure 21(b): load/process after spawning one instance. As the graph indicates, the number of nodes is indicated as 1; the number of CPUs is indicated as 2; and the number of processes is increased from 2.2 to 11 between the timings 16–16:20. The figure shows the cluster memory metric. After one instance is spawned, cache, buffer, and total memory size are shown in Figure 16. The patient information after a time period gets accumulated, and the number of processes to save and store temporarily is being decided and represented.

Figure 21(c): cluster memory after one instance spawn. This figure shows the CPU usage in terms of user, nice, system, wait, and idle. After one instance is spawned, the number of users' cache, buffer, and total memory size. The group of patient data nodes is collected by the cluster after instance creation.

Figure 21(d): cluster CPU usage after one instance spawn. This figure shows network usage. The entire health network usage is being depicted.

Figure 21(e): cluster network after 1 instance spawn. This figure shows the legend and the cluster network variation after the data received are depicted.

Figure 21(f): load versus time after one instance spawn. It is a sample load pertaining to the hospital server.

Figure 22 represents the time at which another instance is brought up and the resource availability.

Figure 22(a): availability after 2 instances spawn. The figure shows the load/process. The data path availability is shown for data transfer.

Figure 22(b): cluster load after 2 instances spawn. This figure shows the cluster memory usage. After monitoring 2 patient nodes, the memory occupied is depicted.

```

bonny@myueccluster:~$ euca-describe-availability-zones verbose
AVAILABILITYZONE myueccluster 192.168.10.121
AVAILABILITYZONE |- vm types free / max cpu ram disk
AVAILABILITYZONE |- m1.small 0003 / 0004 1 192 2
AVAILABILITYZONE |- c1.medium 0003 / 0004 1 256 5
AVAILABILITYZONE |- m1.large 0001 / 0002 2 512 10
AVAILABILITYZONE |- m1.xlarge 0001 / 0001 2 1024 20
AVAILABILITYZONE |- c1.xlarge 0000 / 0000 4 2048 20
bonny@myueccluster:~$
    
```

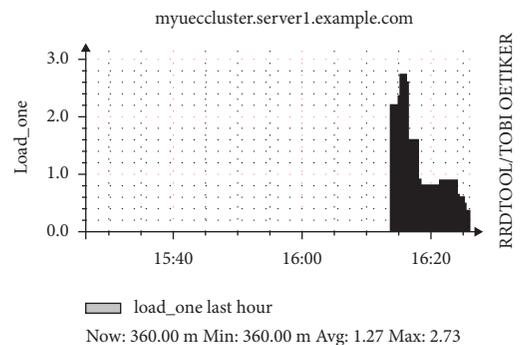
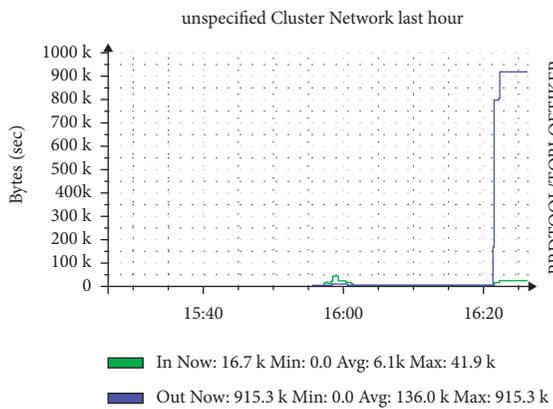
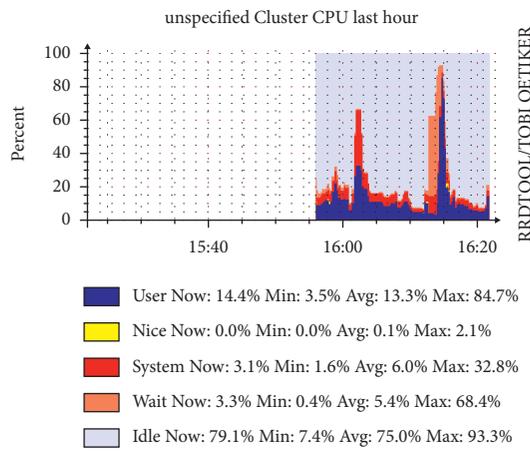
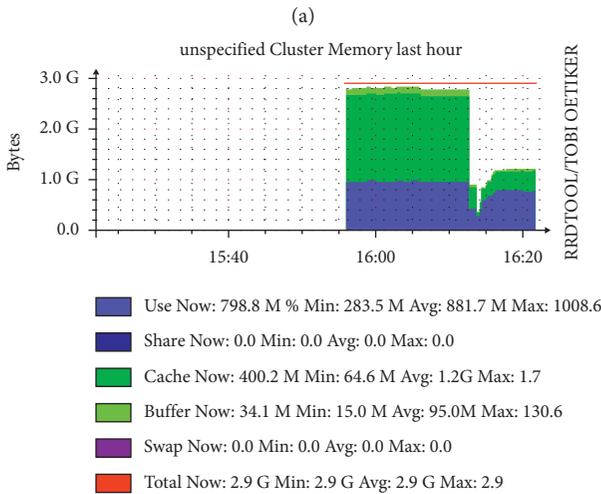
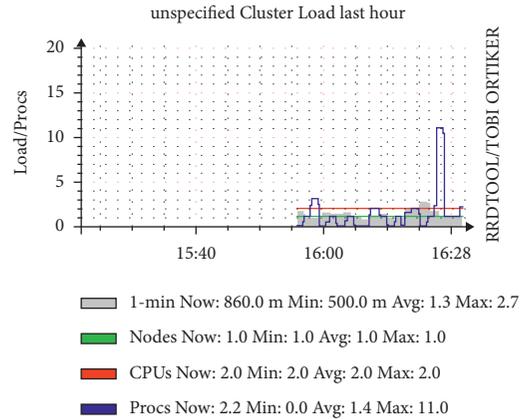


FIGURE 21: Various stages of network resources at a specific time instance.

Figure 22(c): cluster memory after 2 instances spawn. This figure shows CPU usage. After the monitoring of 2 nodes, the CPU processing is depicted.

Figure 22(d): cluster CPU after 2 instance spawn. This figure shows network usage. The cluster CPU, which resides and does the operation over the node data, after acquiring 2 nodes' data is shown in this figure.

Figure 22(e): cluster CPU after 2 instances spawn. This figure shows network usage. The network corresponds to the hospital is depicted with regard to its workload.

Figure 22(f): cluster network after 2 instances spawn. This figure shows the legend. The combination of clusters in a cluster network is depicted.

Figure 23 shows the resources available after the spawn of the third instance. The resources for computing, memory, and storage are exhibited.

```

bonny@myuecluster:~$ euca-describe-availability-zones verbose
AVAILABILITYZONE myuecluster 192.168.10.121
AVAILABILITYZONE | - vm types free / max cpu ram disk
AVAILABILITYZONE | - m1.small 0002 / 0004 1 192 2
AVAILABILITYZONE | - c1.medium 0002 / 0004 1 256 5
AVAILABILITYZONE | - m1.large 0000 / 0002 2 512 10
AVAILABILITYZONE | - m1.xlarge 0000 / 0001 2 1024 20
AVAILABILITYZONE | - c1.xlarge 0000 / 0000 4 2048 20
bonny@myuecluster:~$
    
```

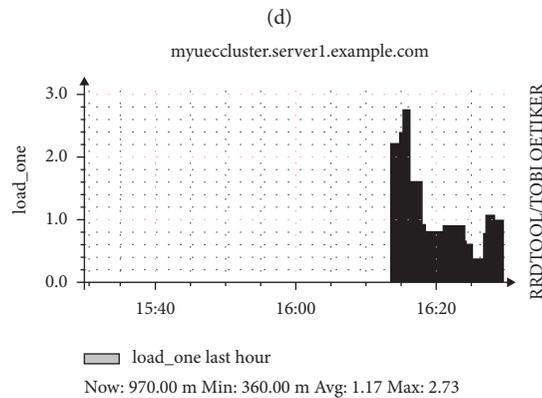
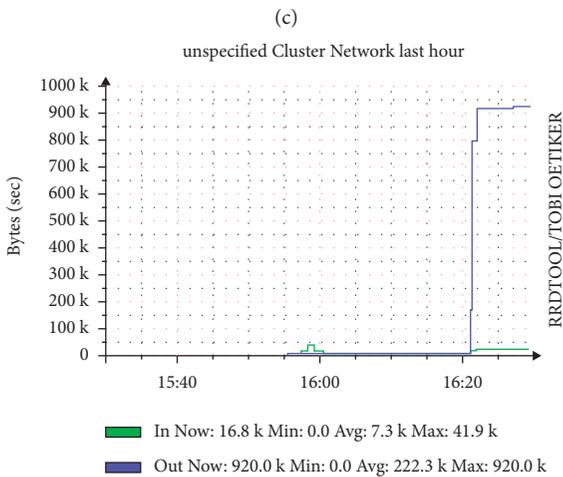
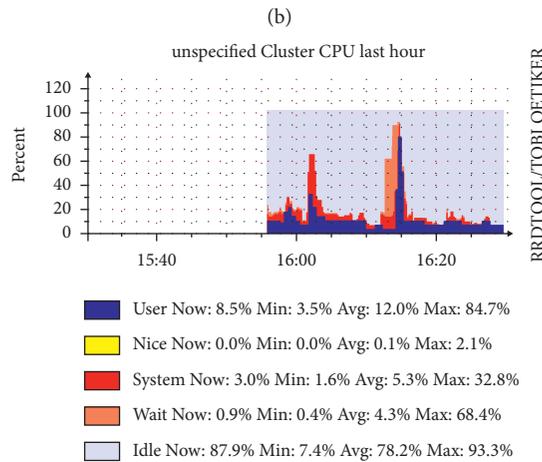
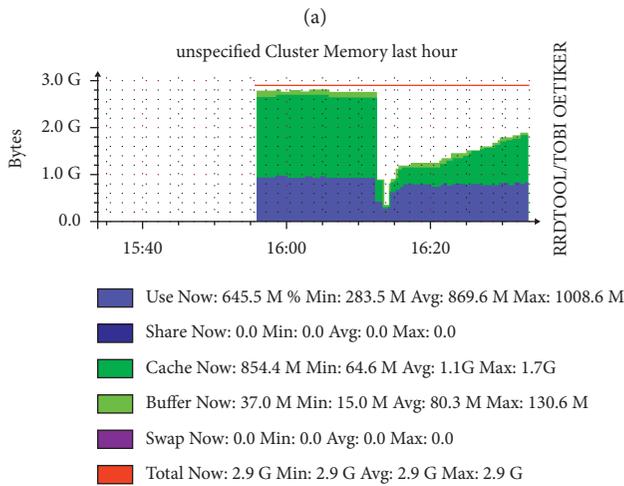
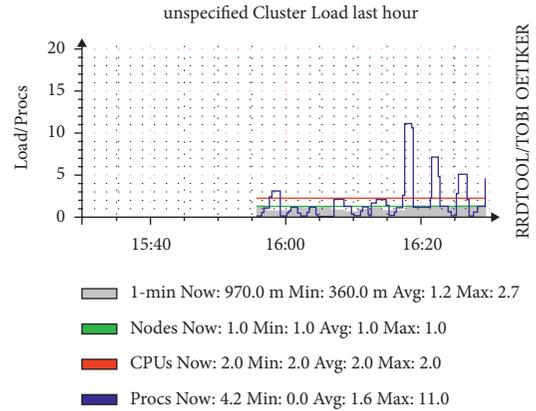


FIGURE 22: Time at which another instance is brought up and the resource availability.

Figure 23(a): the resource usage. This figure shows the load/process. The usage of loading of a patient data per process is depicted.

Figure 23(b): cluster load after 3 instances spawn. This figure shows the cluster memory usage. The cluster is a collection of patient nodes; the cluster load is exhibited.

Figure 23(c): cluster memory after 3 instances spawn. This figure shows CPU usage. The processes acquiring each patient information after the 3 instances of patient node are being depicted.

Figure 23(d): cluster CPU after 3 instances spawn. This figure shows network usage. The cluster CPU resides

```

bonny@myueccluster:~$ euca-describe-availability-zones verbose
AVAILABILITYZONE myueccluster 192.168.10.121
AVAILABILITYZONE | - vm types free / max cpu ram disk
AVAILABILITYZONE | - m1.small 0001 / 0004 1 192 2
AVAILABILITYZONE | - c1.medium 0001 / 0004 1 256 5
AVAILABILITYZONE | - m1.large 0000 / 0002 2 512 10
AVAILABILITYZONE | - m1.xlarge 0000 / 0001 2 1024 20
AVAILABILITYZONE | - c1.xlarge 0000 / 0000 4 2048 20
bonny@myueccluster:~$
    
```

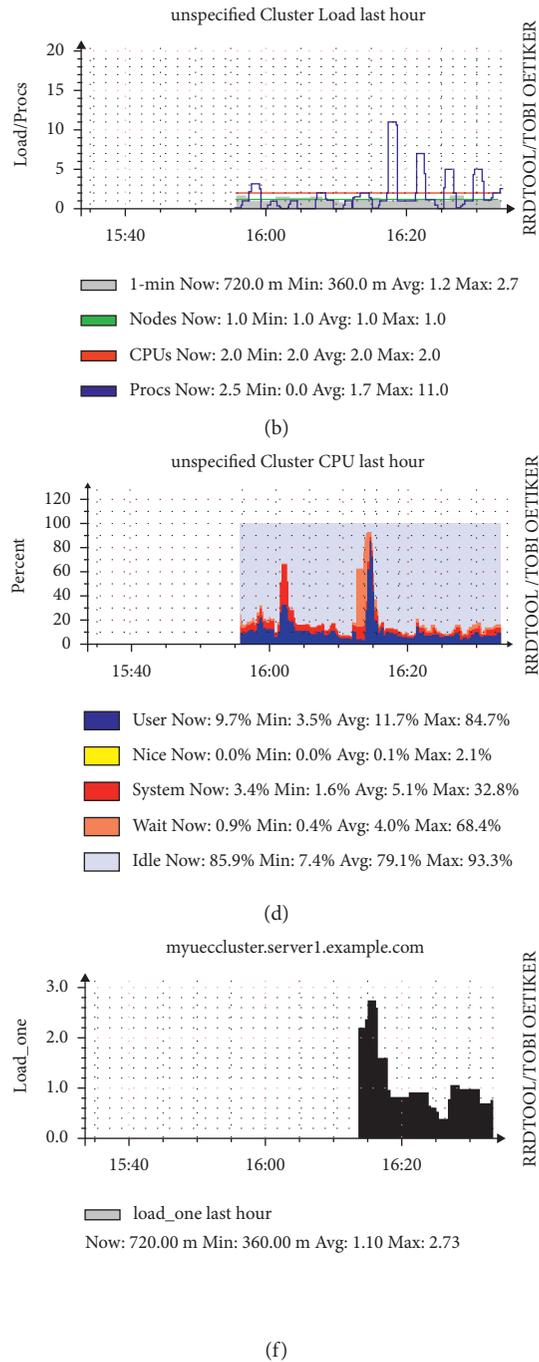
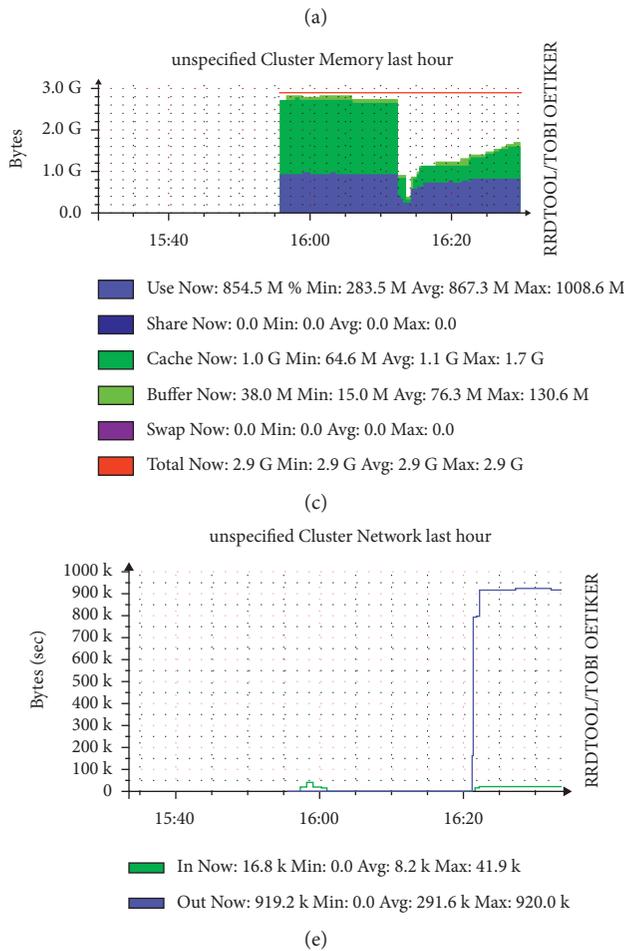


FIGURE 23: Availability zone after 3 instances spawn.

and does the operation over the node data, after acquiring 3 nodes' data, which is shown in the figure.

Figure 23(e): cluster network after 3 instances spawn. This figure shows the legend. In the collection of clusters, the entire health network after the 3 instances is depicted.

Figure 23(f): load after 3 instances spawn. This figure shows the load on that time instant. Cluster load on a particular time instant is depicted.

Figure 24 represents the last available instance, spawned up with the availability zone that shows the resources.

Figure 24(a): cluster load after 4 instances spawn. This figure shows the load/process. When the number of clusters has been increased to 4, the cluster load is depicted.

Figure 24(b): cluster memory after 4 instances spawn. This figure shows the cluster memory. The memory after 4 instances of nodes is depicted.

```

bonny@myueccluster:~$ euca-describe-availability-zones verbose
AVAILABILITYZONE    myueccluster    192.168.10.121
AVAILABILITYZONE    |- vm types     free / max    cpu  ram  disk
AVAILABILITYZONE    |- m1.small     0000 / 0004  1   192  2
AVAILABILITYZONE    |- c1.medium    0000 / 0004  1   256  5
AVAILABILITYZONE    |- m1.large     0000 / 0002  2   512  10
AVAILABILITYZONE    |- m1.xlarge    0000 / 0001  2  1024  20
AVAILABILITYZONE    |- c1.xlarge    0000 / 0000  4  2048  20
bonny@myueccluster:~$
    
```

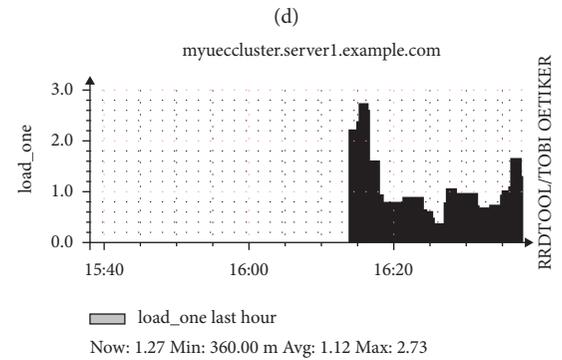
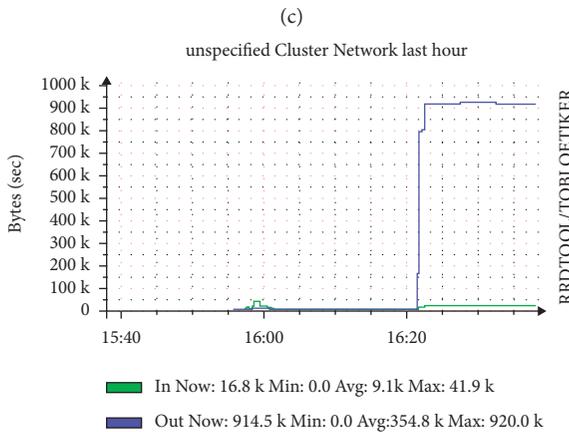
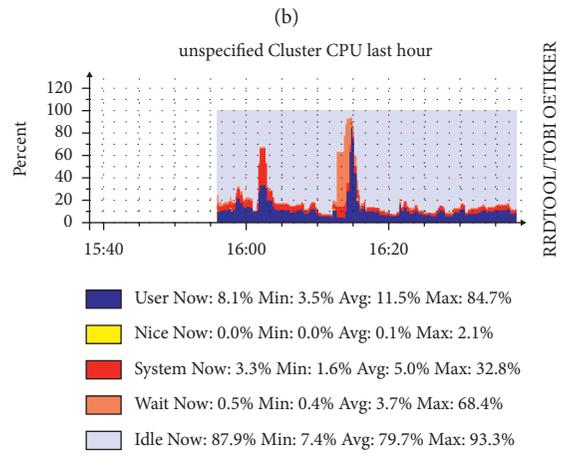
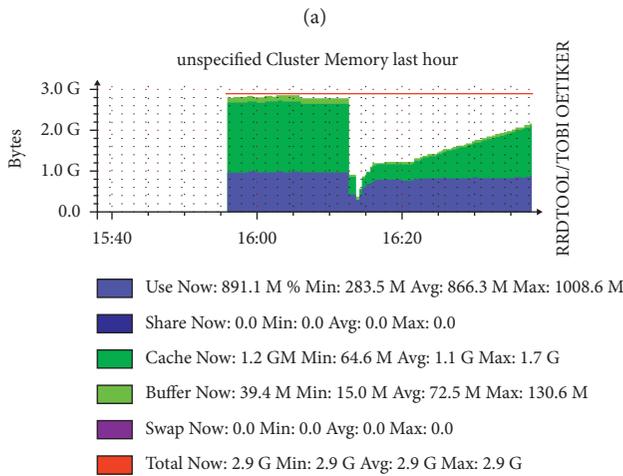
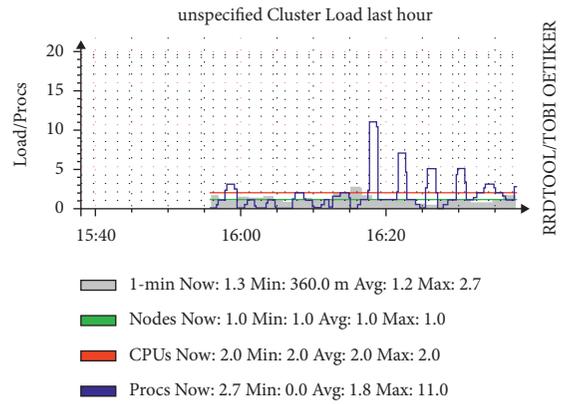


FIGURE 24: The resources are represented by the last available instance, which was generated with the availability zone.

Figure 24(c): cluster CPU after 4 instances start. This figure shows CPU usage. After 4 instances, the CPU utilization is being depicted.

Figure 24(d): cluster network after 4 instances spawn. It shows the cluster network usage and the efficiency in scalability.

Figure 24(e) shows legend after 4 instances spawn. This figure shows the load one metric after all instances are spawn.

Figure 25 shows the network traffic cover the cloud set up. If another instance is to be spawned up, the error that occurs is shown with no availability of resources.

The system information about the cluster is shown in Figure 26.

Figure 27 shows the metrics for load, CPU, memory, and network.

Figure 28 shows the memory metrics. It is based on memory buffers, cached memory, free memory, shared memory, and free swap space.

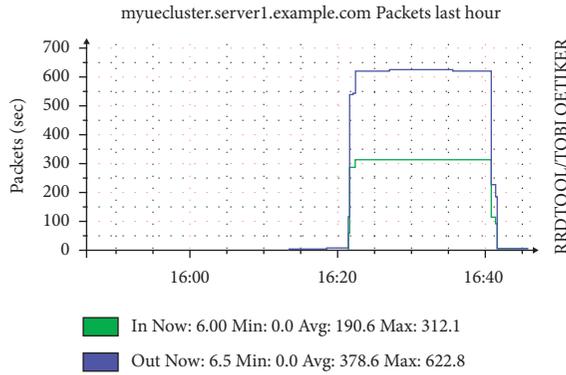


FIGURE 25: Packets/second in the network.

CPU's Total: 2
 Hosts up: 1
 Hosts down: 0

Current Load Avg (15, 5, 1m):
 38%, 52%, 60%

Avg Utilization (last hour):
 58%



Cluster Load Percentages
 50-75 (100.00%)

FIGURE 26: Cluster load percentages.

The network metrics are based on bytes received, bytes sent, packets received, and packets sent, which is shown in Figure 29.

Figure 30 shows the CPU metrics based on CPU idle, CPU idle, CPU Nice, CPU system, CPU user, and CPU wio.

The process metrics are based on total running processes, and the number of total processes is shown in Figure 31.

7. Interpretation of Results

The study of the above graphs makes it all very clear. The cloud is an option for data scalability for handling and analyzing it efficiently and can be used for the same till the resources hit the threshold value of the resources in the cloud. It can be seen from the graphs that there is a steep rise in load as the instances are brought up. The memory usage also rises linearly at an observable high slope value. The network usage steps up at the spawning up of the first instance remains fairly constant thereafter. Bottlenecks can be observed mostly in cluster memory. The CPU bottleneck problem may start if too many complex operations are performed in the cluster. The load on the cloud increases linearly at a fairly low slope value.

Hadoop MapReduce is yet to be implemented on the Eucalyptus cluster, and a study is made on the performance. The same scenario shall be tested on the OpenStack cloud platform, and a comparison shall be made based on the

results with Eucalyptus. Security has not been enforced at the enterprise level. There are myriad works to be done using the cloud and shall be published soon. Considering data availability, when one instance has come many instances, then many servers may have the information at many locations. The locations of the servers were the physician's locations correlating the prioritized instance generation and management mechanism.

7.1. Performance Metrics. Performance of the proposed system can be measured in terms of throughput, bandwidth, delay/latency, packet delivery ratio, signal-to-noise-interference ratio, link quality, nice, buffer size, cache hits, heterogeneity of packets, relationship between group-related instances that can be applied interchangeably with the number of instances and number of packets transmitted generated, relationship between packet loss and SINR and link failure, and scaling due to buffer size increment and path optimization as shown in Table 2.

If $0\% < \text{CPU usage} > 80\%$, redirect the virtual instance, else assign a new CPU—when to provide what type of resource with what capacity?

If $0\% < \text{Memory usage} > 80\%$, increase buffer size, else assign external memory—path optimization and flush out cache memory using a specific set of algorithms.

If $0\% < \text{Network usage} > 80\%$, redirect the virtual instances in an unused path, else assign a new path closer to the destination—load balancing and prioritized packet transfer.

If $0\% < \text{Cluster usage} > 80\%$, redirect the virtual instance towards destination, else assign a new cluster—apply network partitioning and specify a new topology to handle critical instances.

If $0\% < \text{load/process time} > 80\%$, reduce the load by reassigning any of the four mentioned categories—apply utilization factor estimations and time-out mechanisms.

7.2. Software-Defined Policies for Smart Health

7.2.1. Software-Defined Policy for Resource Allocation. Resources offered in a global data network can be as follows:

- (1) CPU
- (2) Memory
- (3) Cluster
- (4) Network
- (5) Grid

The process parameters required can be as follows:

- (1) Process/load time
- (2) Throughput/packet delivery ratio (packets transmitted, received, and dropped)
- (3) Latency/delay/jitter
- (4) Congestion/network traffic
- (5) Link quality/SNR
- (6) Utilization factor/buffer size/BER

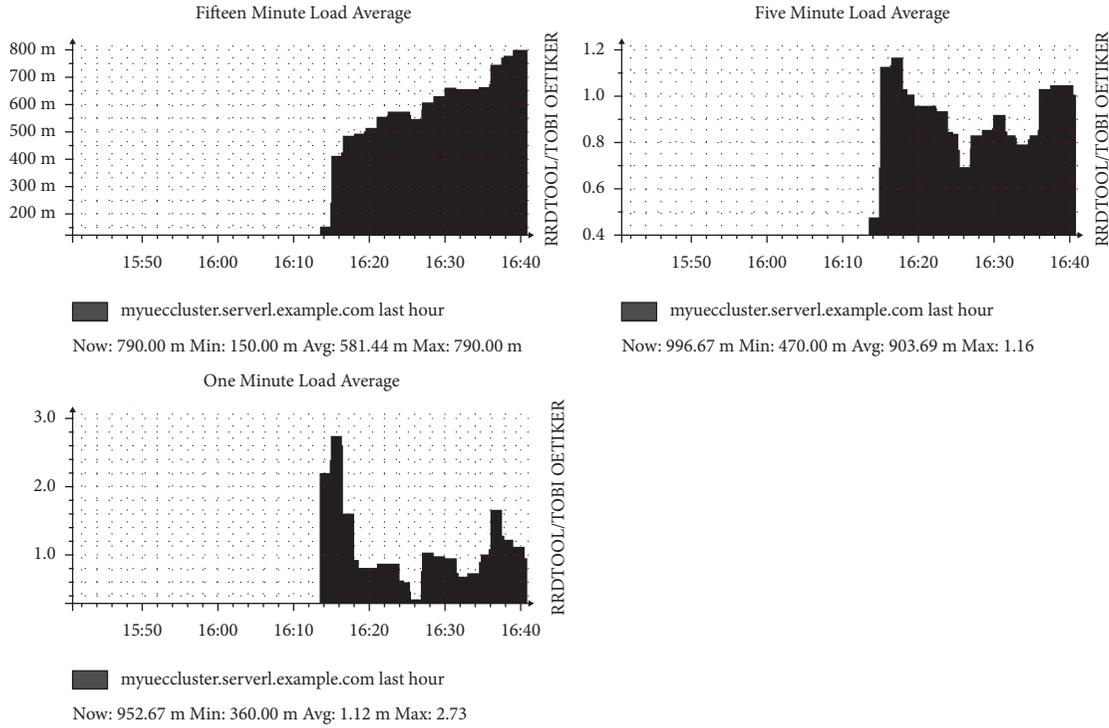


FIGURE 27: Load metrics over time.

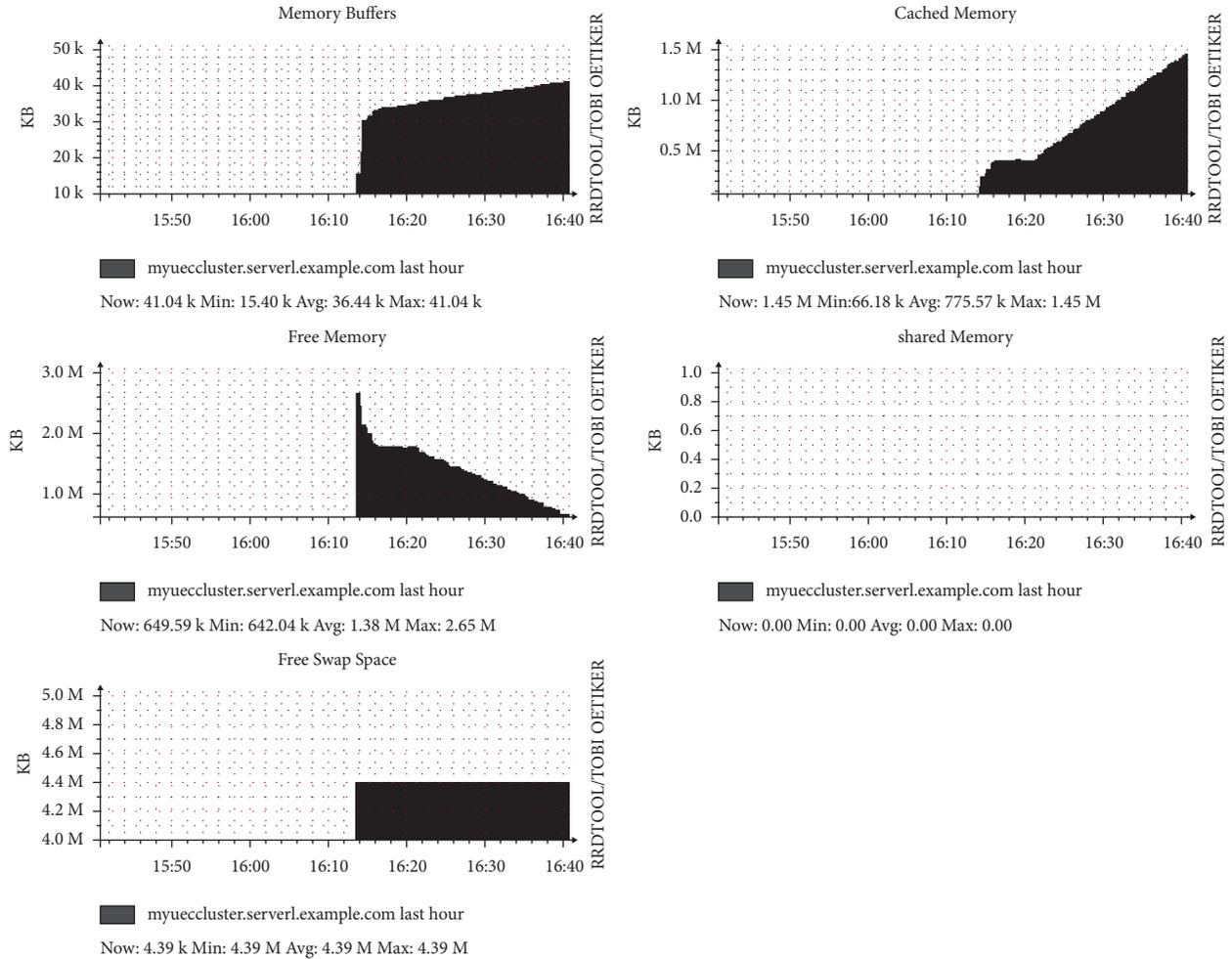


FIGURE 28: Memory metrics.

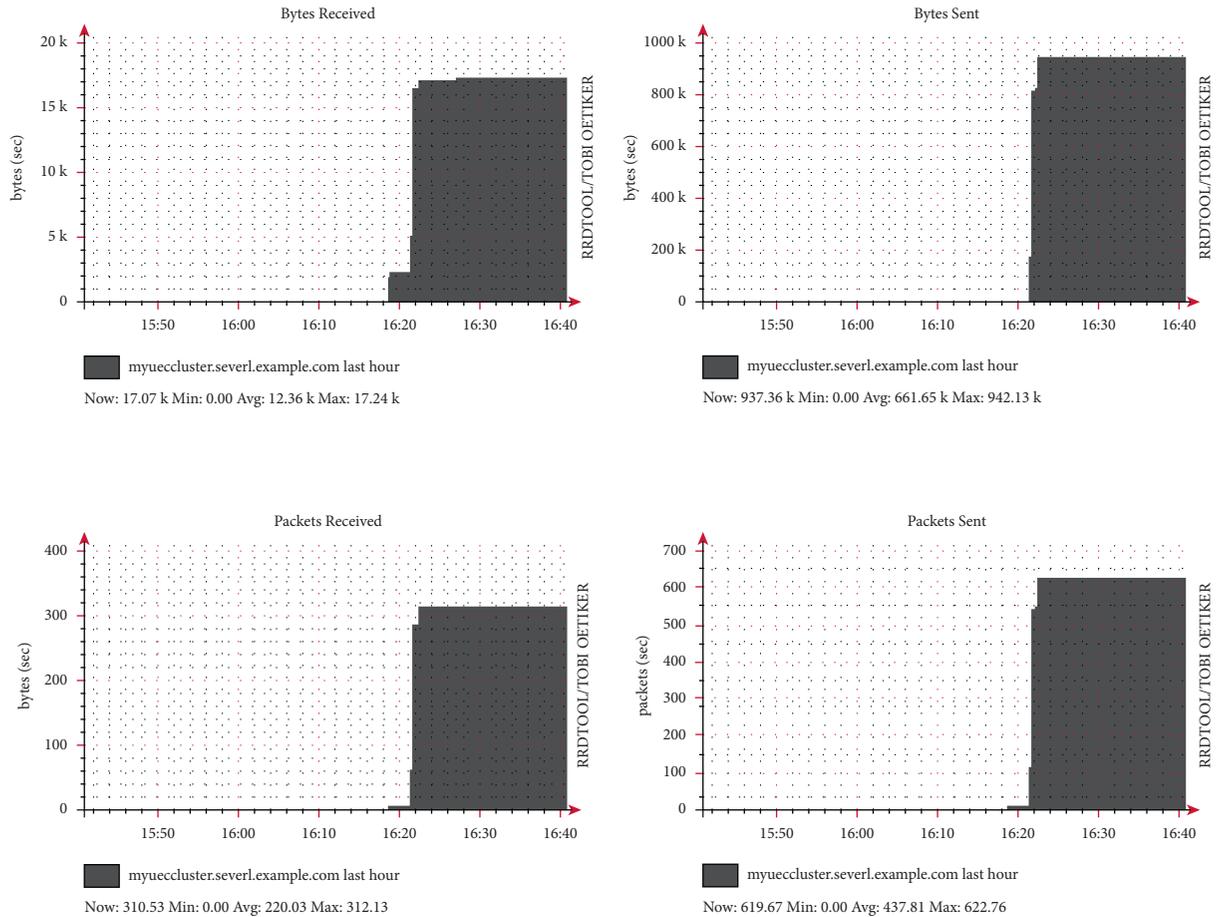


FIGURE 29: Network metrics.

- (7) Software-defined policy for the smart healthcare application was based on the three mentioned categories
- (8) Parameter-/data-centric—high, medium, normal, and low
- (9) Network-centric—topology, flow, and routing protocol
- (10) Node-centric/IP address—location based and distance-based (hop distance)

(1) *Case I: Parameter-/Data-Centric.* If the healthcare data relating to criticality is of high value, it is mentioned as CF_{IM} (critical factor – immediate), which lies on or nearer to both the extremities of death condition values. For this kind of scenario, emergency path allocation was done.

If the healthcare data relating to criticality is of medium value, it is declared as CF_{med} (critical factor – medium), which lies in the medium range of the disease spectrum, where the diagnosis can be done at a moderate level. For this kind of scenario, a congestion-free path can be allocated through the network.

If the healthcare data linking to criticality is of normal value, it is stated as CF_{nor} (critical factor – default), where continuous patient monitoring is required. For this kind of

scenario, the data was allowed to flow through the paths with the least congestion/congestion-free paths to reach the destination.

If the healthcare data relating to criticality is of low value, it is revealed as CF_{low} (critical factor – low), which means these types of patients can wait for a while to meet the doctor for a diagnosis. For this kind of scenario, the data was allowed to flow through the available paths/less-congested paths with an average waiting time of patients in the hospital for the diagnosis.

(2) *Case II: Network-Centric.* Analyzing the network characteristic, the data have to reach the destination in time based upon the following three factors:

- (1) Topology-based: response time
- (2) Flow-based: ingress/egress
- (3) Routing-protocol-based: proactive/reactive

Topology-based: response time. The healthcare data that is fed as an input to a system connected through any topology, say star, mesh, ring, and so on, has to be forwarded to the destination that is the nearest doctor or hospital based on availability. A dynamic software mechanism can be built in the forwarding/routing/gateway node, so the data can reach the

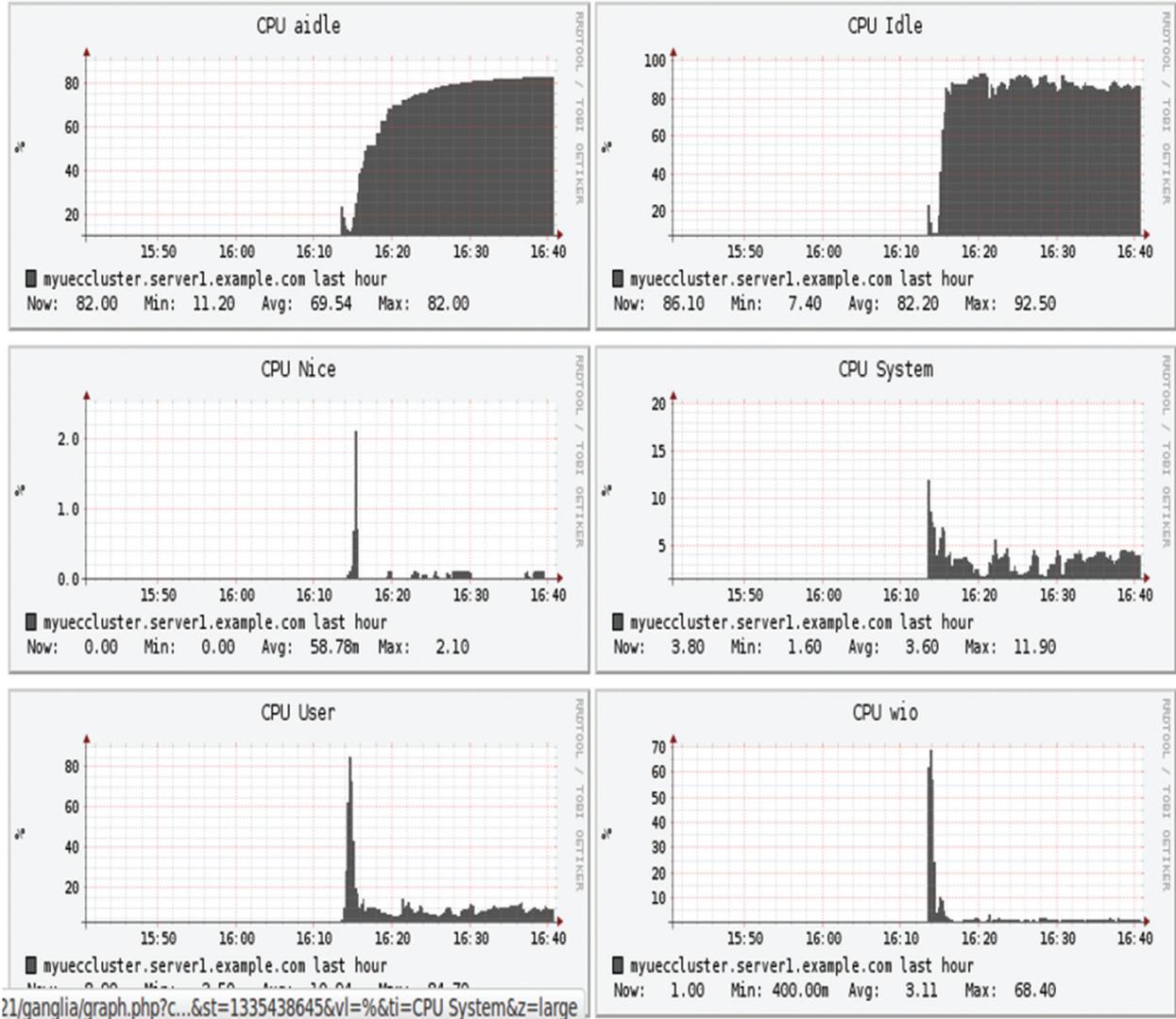


FIGURE 30: CPU metrics.

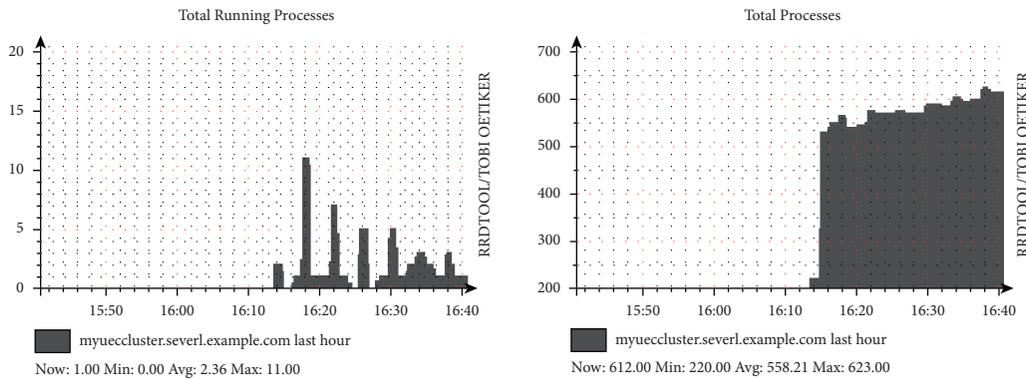


FIGURE 31: Process metrics.

TABLE 2: Sample data requirements for resource allocation.

Number of instances	Packets transmitted	Packets received	Packet loss	Packet delivery ratio/throughput	Signal-to-noise interference ratio (dB)	Network delay (ns)	BER	Resource needed
1,20,000	120	118	02	98.33	98	0.002	0.67	CPU

destination, based on response time requirements. When the topology change occurs, if the data from the forwarding node can reach the destination node at a lesser time than the previously connected topology, the new topology change has to be accomplished.

Flow-based: ingress/egress. Depending upon the inflows and outflows of a particular node or a system, the waiting time of a healthcare data packet is made nil. For the flow to be processed, according to the flow table entry, the actual number of servers in the up condition was identified using the link quality factor, and then the flow was forwarded towards the responsive server.

Routing-table-based: the routing protocol. It can be used for transferring the patient data is categorized typically as proactive and reactive.

Under proactive protocol-based routing, the patient data that are under critical conditions can be transferred, and the patients under continuous monitoring can also be transferred for doctor's diagnosis at any time.

Under the reactive protocol, if a patient's cumulative healthcare data for a specific time is required for analysis by the expertise, the router will forward the historical data for diagnosis.

(3) *Case III: Node-Centric (IP Address)*. TCP/IP provides end-to-end connectivity. Node-centric data avoid the communication overhead by combining the features of TCP/IP specification of how data can be packetized, addressed, transmitted, routed, and received at the destination. A communication network should allow a user to focus on the data he or she needs, wherein smart healthcare application, the ultimate aim is to provide the required healthcare to any patient at any time regardless of the geographical location, the patient is situated or met an accident or in trauma. Therefore, to refer to a specific, physical location where that data is to be retrieved is required. Node-centric data networking comes along with the concept of TCP/IP such as data caching to reduce congestion and improve delivery speed, simpler configuration of network devices, and building security into the network at the data level shown in Figure 32.

Communication in node-centric networking is driven by receivers at the hospital side through the exchange of two types of information packets: IP address and patient data. Both types of packets carry a location name that can be transmitted in a single data packet.

IP address (source): the patient data is kept in a data packet along with the IP address of the node that is the creator and transfers the data. This data is embedded into an IP packet that is then pushed to flow in the network towards the destination using a congestion-free path based on the response time requirement of the packet.

Destination: once the embedded packet reaches the destination node in the nearby hospital, based on

proactive or reactive mode, the node will return an acknowledgment that contains both the IP address and the patient data packet along with a signature by the destination's IP address, which binds the two. This packet follows in reverse the path taken by the IP addressed data interest to get back to the source.

7.2.2. Software-Defined Policy on Process Parameter Requirement

- (1) Process/load time: based on the number of instances, considering data availability and scalability issues,

$$\sum_{i=1}^n \frac{e_i}{p_i} = \sum_{i=1}^n u_i \leq 1. \quad (23)$$

A task set is schedulable under critical instances; e_i is the execution time of the task in the processor based on its specification. p_i is the period of task, and u_i is the utilization factor. The earliest deadline first algorithm is modified with the criticality and the emergency of the patient, which will suitably take up the process can be preferable.

- (2) Bandwidth demands: a real-time signal processing application like healthcare data processing computes in each sampling period one or more outputs. Each output $X(k)$ is a weighted sum of n inputs $Y(i)$, expressed as follows:

$$X(k) = \sum_{i=1}^n a(k, i)Y(i). \quad (24)$$

The weights $a(k, i)$ are known as per prioritization of the data and fixed. The processor time-bound is also measured based on the response time requirement of producing a specific number of outputs in each sampling period.

- (3) Execution time: execution time is the amount of time required for the processor to complete the execution of the task J_i when it executes alone and has all the resources it requires.
- (4) Precedence constraints: AND/OR precedence constraints can be evaluated to combine the simultaneous critical events at the same location, where the number of emergency cases is more.
- (5) Usefulness function: the usefulness function is used to describe qualitatively the real-time performance objectives of the smart healthcare system. It emphasizes how a single patient can feel concerned about his/her medical diagnosis. The usefulness function can guide the choice and implementation of the scheduling strategies.
- (6) Lateness: lateness is defined as the difference between its completion time and its deadline mentioned to complete that is the response time of the patient data to be attended.
- (7) Priority driven approach for instance execution: the patient data is continuously perceived through the

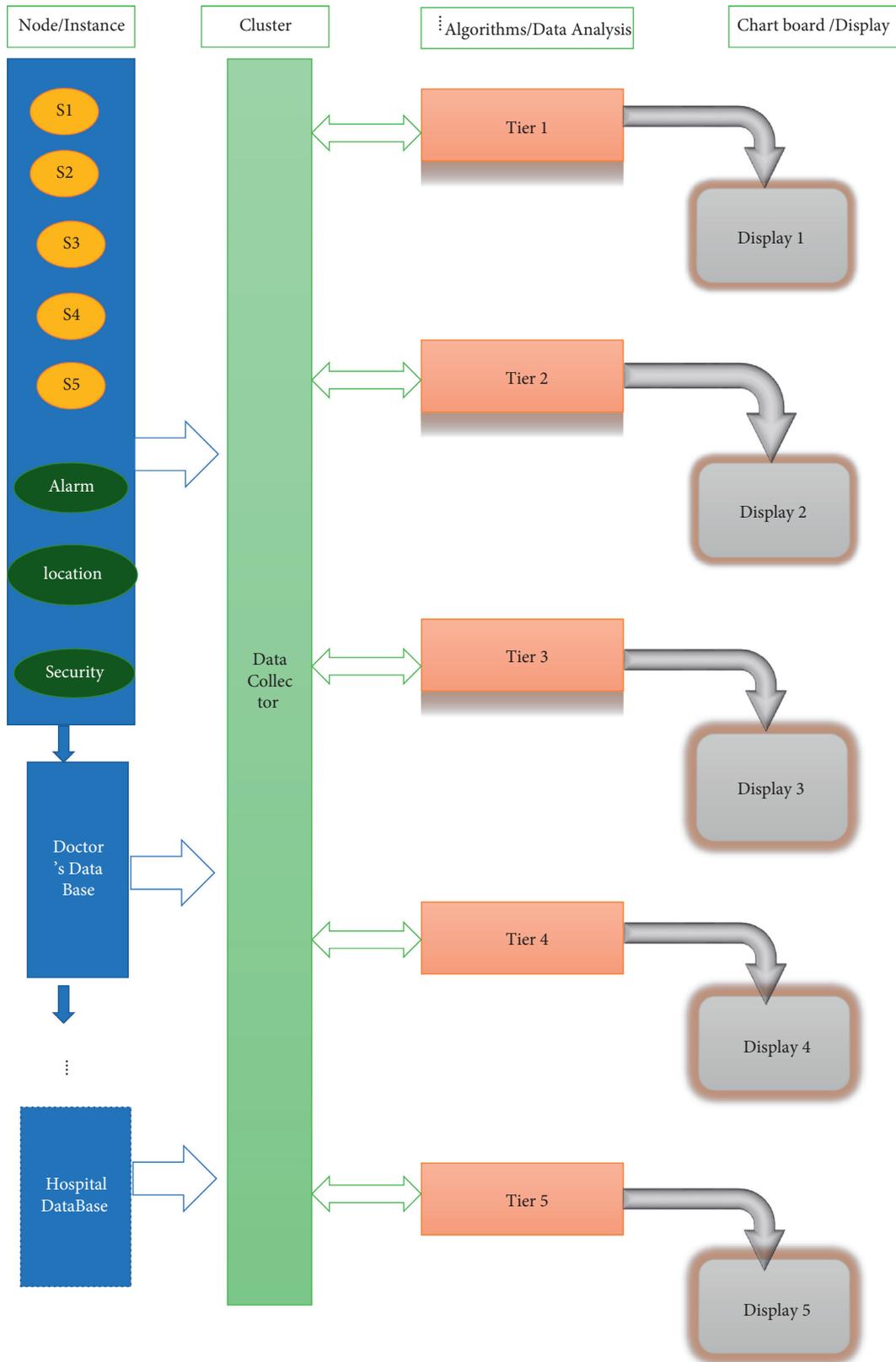


FIGURE 32: Node-centric data.

sensors and the streaming platform streams the patient data. As the patient data has been prioritized and categorized as normal, abnormal, and critical, the simulation environment must create instances appropriately. The patient's data collector has been considered as a single node. The Amazon EC2 console launches an instance for each node, which will be a part of the cluster. Also, the workload for the single cluster is required, based on the number of nodes, as the nodes specify the number of instances, against the cluster, a separate instance for that workload is also created. To know the survivability of the clusters, minimum of three nodes have been run under trial. The instances rely on Amazon Time Sync Service for the clock synchronization among the nodes. When choosing an AMI, some of the machines are preconfigured to use Amazon Time Sync Service, whereas the other machines are not preconfigured.

There are three instance types used such as m for general purpose, c for compute-optimized, and i for storage-optimized, with instance store volumes. We can have, for example, 16 vCPUs and 32 GB of RAM per instance, for internal testing depending upon our workload. When creating an instance for the node, a private key file must be downloaded to be used to securely connect to the instance. The location of the file must be decided and the file path also.

To configure our network, the custom TCP inbound rules to the security group must be enabled to allow TCP communication on two ports for inter- and client-node communication. This task enables the nodes to work as a cluster and make the load balancer route the traffic to the nodes and the healthcare application to connect to the load balancer. A procedure to timer activation, instance creation, network instigation, security mechanisms, and triggering the entire operation through simulation is as follows.

7.2.3. Procedure

- (1) Activate timer
- (2) Instances remaining = 1000; //Observe the number of instances at a specific time period
- (3) Next release time = processor clock + 100 units; // Current processor time at source + required response time //Arrival time at the destination
- (4) While (instances remaining > 0)
- (5) Now = processor clock
- (6) If (now < next release time), do
- (7) Timer sleep until (next release time)
- (8) Instances waiting in the specific type of task; //based on the criticality of the data such as high, medium, normal, and low
- (9) Next release time = next release time + 100 units
- (10) Else

- (11) Instances in the program of the aperiodic task; //Any patient data can be injected into the network at any time
- (12) Next release time = Now + 100 units
- (13) Instances remaining = instances Remaining - 1
- (14) End while
- (15) Thread-destroy (thread-ID)

7.3. Advantages of Ganglia in Performance Monitoring. A clustering coefficient quantifies the degree to which nodes in a graph cluster. To create tightly knit groups with a relatively high density of ties within the WBAN network. The probability of ties is greater than the average probability of establishing a tie randomly between two nodes [38, 39]. There are two types of tie connectivity between nodes: global and local.

The global version was created to provide an overview of the network's clustering, whereas the local version provides information about the embeddedness of individual nodes within WBAN networks. Both are useful in terms of performance dimensions.

The global clustering coefficient is computed using node triplets. Three nodes are connected by either two or three undirected ties to form a triplet. The global clustering coefficient is defined as the ratio of closed triplets to total triplets. This metric indicates network clustering and can be applied to both undirected and directed networks, both of which are referred to as transitivity [42, 43].

The global clustering coefficient is defined as follows:

$$C = \frac{3 \times \text{number of triangles}}{\text{number of connected triplets of vertices}} \quad (25)$$

$$= \frac{\text{number of closed triplets}}{\text{number of connected triplets of vertices}}.$$

Each triangle forms three connected triplets. A generalization to weighted networks was proposed [44], and a redefinition to two-mode networks was expressed after that [45].

The local clustering coefficient of a node in a network shows looseness towards its neighborhood to form a clique. Such a graph can be treated as a small-world network.

A graph $G = (V, E)$ formally consists of a set of vertices V and a set of edges E between them. An edge represented as e_{ij} connects vertex v_i with vertex v_j .

The neighborhood N_i for a vertex v_i can be represented by its adjacently connected neighbors as follows:

$$N_i = \{v_j, e_{ij} \in E \cap e_{ji} \in E\}. \quad (26)$$

As e_{ij} and e_{ji} indicates an undirected graph, K_i can be defined as the number of vertices, $|N_i|$, in the neighborhood, and N_i , of a vertex.

The local clustering coefficient C_i for a vertex v_i is given by the proportion of links between the vertices within its neighborhood divided by the number of links that could exist between them [40].

An undirected graph has the property that e_{ij} and e_{ji} are considered identical. Therefore, if a vertex v_i has K_i neighbors, $(k_i(k_i - 1)/2)$ edges could exist among the vertices within the neighborhood. Thus, the local clustering coefficient for undirected graphs can be defined as follows:

$$C_i = \frac{2|\{e_{jk}: v_j, v_k \in N_i, e_{jk} \in E\}|}{k_i(k_i - 1)}. \quad (27)$$

7.3.1. Network Average Clustering Coefficient. In addition to the global clustering coefficient, Watts and Strogatz define the overall level of clustering in a network as the average of the local clustering coefficients of all the vertices n .

$$\bar{C} = \frac{1}{n} \sum_{i=1}^n C_i. \quad (28)$$

This metric favors low degree nodes, whereas the transitivity ratio favors high degree nodes. A weighted average is identical to the global clustering coefficient when each local clustering score is weighted by

$$k_i(k_i - 1). \quad (29)$$

7.4. Case Study: A Sample Smart Health Network. Figure 33 presents the sample network that exists between the patients and the hospitals. S1, S2, . . . , S5 indicates the available patients at different locations, and they are not interconnected with each other. D1, D2, . . . , D5 indicates the availability of hospitals/doctors in predefined locations spreading geographically. The sample network connects both the patients and the hospitals through the existing paths. The available paths are mentioned as follows without considering the network congestion.

Based on the number of existing paths and available paths, patient data have to reach the destination within the mentioned response time. The patient data have to be collected from the patient using a data collector, and from the data collector, it reaches the fog node. From the fog node, it enters into the Internet to the cloud for further processing. The data transfer time between any two nodes depends upon the distance and the speed of data transfer between the nodes in the hospital network. In the simulation, the following procedure has been adopted.

- Step 1. Create instances
- Step 2. Configure the hospital network—assigning the nodes
- Step 3. Synchronize clocks of the nodes in the network
- Step 4. Set up load balancing as the data can flow in any available path
- Step 5. Generate certificates for the nodes
- Step 6. Start nodes to function
- Step 7. Initialize the cluster, which comprises a set of nodes
- Step 8. Testing and configuring the cluster with all the nodes

- Step 9. Run a sample workload using the instances
- Step 10. Monitor the cluster for its performance
- Step 11. Scale the cluster for more number of nodes to participate
- Step 12. Use the database to store and for further compute operations
- Step 13. Estimate the bandwidth for the service and fix the data transfer speed
- Step 14. Consider the data size and ensure the reachability of data

To analyze the performance of the system, the following table exhibits the constraints behind it. The type of topology and congestion in the path are not now considered in the table shown in Table 3.

In Table 4, there is only one path available for the sources and destinations, regardless of the response time requirement, the only existing path that has to be chosen for data transfer, whereas if more than one path exists between the source and the destination, based on the response time requirement, whichever path is having less travel time was chosen for data transfer. If more than one path is having similar travel times, any path that is having less congestion was chosen for data transfer. These constraints were made as policies, which were fed into the software-defined controller to route the data packets based on the criticality of the event at the source. The number of virtual instances indicates that the same data packet can be delivered over many paths, with the request being served by whatever path is active at the time (Figure 34). The number of virtual instances to be created was estimated, based on the actual number of events and the weighting factor of the event. Weighting factor values are related to the emergency of the patient's health condition.

8. Concluding Remarks: Salient Features of the Proposed Cyber-Physical System

The WBAN architecture has been devised in terms of the five-tier architecture, and the scalability of extending the hospital network between interhospital service paradigms over the networking environment has been simulated with required diagrams. The solution we proposed provided the opportunity to interconnect not only doctors to the patients regardless of their locations but also interconnect the hospitals regardless of their geographic locations. There are specific algorithms devised to operate at every tier, and the simulation has been executed using Amazon creating the required instances of EC2 so as to know the performance and the scalability measures. The results show evident information about the data transfer mechanisms at normal and abnormal conditions, based on priority and ranking based on emergency conditions.

The performance factors have been estimated in a promised inexpensive, unobtrusive, and unsupervised monitoring method in normal and abnormal patient conditions. The technology fuses the cyber and physical phase of the system that is ubiquitous and affordable, and the

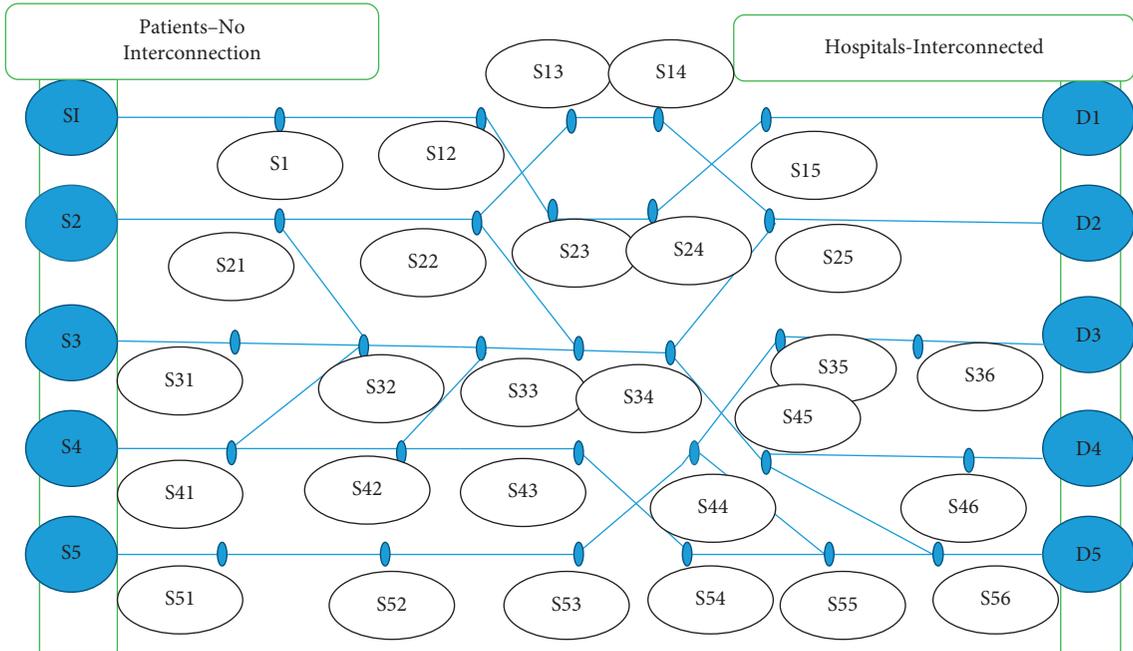


FIGURE 33: Sample network for smart healthcare application.

TABLE 3: Existing path in the sample network between patients and hospitals.

From patient to hospital	Existing paths	Travel time in units
S1-D1	S1—{S11—S12—S23—S24—S15}—D1	0.06
S1-D2	No path	—
S1-D3	No path	—
S1-D4	No path	—
S1-D5	No path	—
S2-D1	No path	—
S2-D2	S2—{S21—S22—S13—S14—S25}—D2 S2—{S21—S32—S33—S34—S25}—D2 S2—{S21—S22—S34—S25}—D2	0.06 0.06 0.06
S2-D3	No path	—
S2-D4	S2—{S21—S32—S33—S34—S45—S56}—D5 S2—{S21—S22—S34—S33—S32—S41—S42—S43—S54—S55—S56} D5	0.07 0.12
S2-D5	S2—{S21—S32—S33—S34—S45—S56}—D5 S2—{S21—S32—S41—S42—S43—S54—S55—S56}—D5	0.07 0.09
S3-D1	No path	—
S3-D2	S3—{S31—S32—S33—S35—S25}—D2 S3—{S31—S32—S21—S22—S13—S14—S25}—D2	0.06 0.08
S3-D3	No path	—
S3-D4	S3—{S31—S32—S33—S34—S45—S46}—D4	0.07
S3-D5	S3—{S31—S32—S33—S34—S45—S56}—D5	0.07
S4-D1	No path	—
S4-D2	S4—{S41—S32—S21—S22—S13—S14—S25}—D2 S4—{S41—S32—S33—S34—S25}—D2	0.08 0.08
S4-D3	No path	—
S4-D4	S4—{S41—S42—S33—S34—S45—S46}—D4	0.07
S4-D5	S4—{S41—S42—S43—S54—S55—S56}—D5	0.07
S5-D1	No path	—
S5-D2	No path	—
S5-D3	S5—{S51—S52—S53—S44—S35—S36}—D3	0.07
S5-D4	No path	—
S5-D5	S5—{S51—S52—S53—S44—S55—S56}—D5	0.07

TABLE 4: Chosen path as per response time requirement.

Type of event (TE)	Number of events (E_i)	Weighting factor (W_i)	Number of actual instances = $\sum_{i=0}^n (A_i) = E_i W_i$	Number of existing paths from source to destination ($S_p - D_p$)	Number of virtual instances (V_i)	Number of available paths (A_p)	Source-destination (S-D)	Required response time (R_i , units)	Chosen path (S-D)
High	10	1	10	1	10	1	S1-D1	0.06	S1-D1
				1	10	1	S5-D5	0.08	S5-D5
Medium	100	0.75	75	3	225	3	S2-D2	0.06	S2-D2 (any)
				1	75	1	S2-D4	0.07	S2-D4
Normal	200	0.5	100	0	0	0	S3-D3	0	NIL
				1	100	1	S3-D4	0.08	S3-D4
Low	50	0.25	12.5	1	125	1	S4-D4	0.07	S4-D4
				1	125	1	S5-D5	0.07	S5-D5

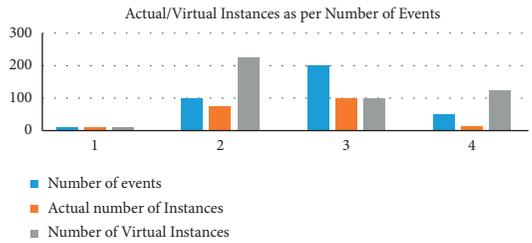


FIGURE 34: Performance analysis as per the number of events.

methodology is a success with the scalability factor also. To make it realizable, the system design, configuration and customization, seamless integration, security and privacy of patient data, and other social issues have to be taken care of.

Fusing physical and cyberspaces through an application is implemented in this research. It can be applied to any domain, where the data usage becomes global. Secure data forwarding is done through the cloud storage system, that is, messages can be forwarded directly between end users over and done with cloud services. Data availability and scalability are maintained using optimization techniques through software control mechanisms. Any patient at any location in the globe can get a timely and proper medical diagnosis through the cyber-physical system architecture. Extensive security and performance analysis are possible from data modification attacks due to the use of encryption using BoxCryptor v2.0 and cloud database storage using Eucalyptus v3.4. User identity management is supported within Eucalyptus with capabilities to control virtual resource pools using fine-grained role-based access control mechanisms for each resource pool. Ganglia supports scalability and automatic resource provisioning techniques with a threshold alert mechanism. This concept can be extended further with the clustering coefficient mechanism. If the rules for physicians are made flexible, to serve any patient anywhere in the world, Ganglia can be used to replicate data across sites [46].

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request (sandeep.kautish@lbef.edu.np).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pp. 88–97, Association for Computing Machinery, Atlanta Georgia USA, 28 September 2002.
- [2] K. Lee and D. Hughes, "System architecture directions for tangible cloud computing," in *Proceedings of the 2010 First ACIS International Symposium on Cryptography, and Network Security, Data Mining and Knowledge Discovery, E-Commerce and Its Applications, and Embedded Systems*, pp. 258–262, IEEE, Suzhou, China, 23 October 2010.
- [3] K. Lee, D. Murray, D. Hughes, and W. Joosen, "Extending sensor networks into the cloud using amazon web services," in *Proceedings of the 2010 IEEE International Conference on Networked Embedded Systems for Enterprise Applications*, pp. 1–7, IEEE, Suzhou, China, 25 November 2010.
- [4] D. Johnson, K. Murari, M. Raju, R. B. Suseendran, and Y. Girikumar, "Eucalyptus beginner's guide-uec edition," Ubuntu Server, 2010 May, <https://fdocuments.in/reader/full/book-eucalyptus-beginners-guide-uec-edition1>.
- [5] R.-G. Lee, K.-C. Chen, C.-C. Hsiao, and C.-L. Tseng, "A mobile care system with alert mechanism," *IEEE Transactions on Information Technology in Biomedicine*, vol. 11, no. 5, pp. 507–517, 2007 Sep 10.
- [6] S. Ivanov, C. Foley, S. Balasubramaniam, and D. Botvich, "Virtual groups for patient WBAN monitoring in medical environments," *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 11, pp. 3238–3246, 2012 Jul 11.
- [7] A. Singh, V. Naik, S. Lal et al., "Improving the efficiency of healthcare delivery system in underdeveloped rural areas," in *Proceedings of the 2011 Third International Conference on Communication Systems and Networks (COMSNETS 2011)*, pp. 1–6, IEEE, Bangalore, India, 4 January 2011.
- [8] I. D. Chakeres and E. M. Belding-Royer, "AODV routing protocol implementation design," in *Proceedings of the 24th International Conference on Distributed Computing Systems Workshops*, pp. 698–703, IEEE, Tokyo, Japan, 23 March 2004.
- [9] Z. Iqbal, M. Gidlund, and J. Åkerberg, "Deterministic and event triggered MAC protocol for industrial wireless networks," in *Proceedings of the 2013 IEEE International Conference on Industrial Technology (ICIT)*, pp. 1252–1259, IEEE, Cape Town, South Africa, 25 February 2013.
- [10] O. Diallo, J. J. P. C. Rodrigues, M. Sene, and J. Niu, "Real-time query processing optimization for cloud-based wireless body area networks," *Information Sciences*, vol. 284, pp. 84–94, 2014 Nov 10.
- [11] S. Movassaghi, M. Abolhasan, and J. Lipman, "A review of routing protocols in wireless body area networks," *Journal of Networks*, vol. 8, 2013 Apr 8.
- [12] H. Zhang, R. Zhang, Q. Li et al., "A GPRS-based wearable electronic thermometric alarm system," in *Proceedings of the 2012 5th International Conference on BioMedical Engineering and Informatics*, pp. 804–807, IEEE, Chongqing, China, 16 October 2012.
- [13] D. P. Tobón, T. H. Falk, and M. Maier, "Context awareness in WBANs: a survey on medical and non-medical applications," *IEEE Wireless Communications*, vol. 20, no. 4, pp. 30–37, 2013 Sep 12.
- [14] B. Bin Liu, Z. Zhisheng Yan, and C. W. Chang Wen Chen, "MAC protocol in wireless body area networks for E-health: challenges and a context-aware design," *IEEE Wireless Communications*, vol. 20, no. 4, pp. 64–72, 2013 Sep 12.
- [15] A. Pantelopoulous and N. G. Bourbakis, "A survey on wearable sensor-based systems for health monitoring and prognosis," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 40, no. 1, pp. 1–2, 2009 Oct 30.
- [16] C. M. Chen, "Web-based remote human pulse monitoring system with intelligent data analysis for home health care," *Expert Systems with Applications*, vol. 38, no. 3, 2011 Mar 1.
- [17] E. Kafeza, D. K. W. Chiu, S. C. Cheung, and M. Kafeza, "Alerts in mobile healthcare applications: requirements and pilot study," *IEEE Transactions on Information Technology in Biomedicine*, vol. 8, no. 2, pp. 173–181, 2004 Jun 7.

- [18] R. K. Megalingam, V. Radhakrishnan, D. C. Jacob, D. K. Unnikrishnan, and A. K. Sudhakaran, "Assistive technology for elders: wireless intelligent healthcare gadget," in *Proceedings of the 2011 IEEE Global Humanitarian Technology Conference*, pp. 296–300, IEEE, Seattle, WA, USA, 30 October 2011.
- [19] R. Cavallari, F. Martelli, R. Rosini, C. Buratti, and R. Verdone, "A survey on wireless body area networks: technologies and design challenges," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, pp. 1635–1657, 2014 Feb 13.
- [20] G. Bhuvaneshwari and A. Rama, "Reliable wireless patient monitoring system using hybrid transmission protocol," *Middle-East Journal of Scientific Research*, vol. 20, no. 4, pp. 468–474, 2014.
- [21] K. Malhi, S. C. Mukhopadhyay, J. Schnepfer, M. Haefke, and H. Ewald, "A zigbee-based wearable physiological parameters monitoring system," *IEEE Sensors Journal*, vol. 12, no. 3, pp. 423–430, 2010 Nov 11.
- [22] J. Aramideh and H. Jelodar, "Application of fuzzy logic for presentation of an expert fuzzy system to diagnose anemia," *Indian Journal of Science and Technology*, vol. 7, no. 7, pp. 933–938, 2014 Jul 1.
- [23] Kim, "A study on the acceptance factor for telehealth service according to health status by group," *Indian Journal of Science and Technology*, vol. 8, no. S1, pp. 542–550, 2015 Jan.
- [24] T. Tamura, I. Mizukura, M. Sekine, and Y. Kimura, "Monitoring and evaluation of blood pressure changes with a home healthcare system," *IEEE Transactions on Information Technology in Biomedicine*, vol. 15, no. 4, pp. 602–607, 2011 May 19.
- [25] R. S. Dilmaghani, Bobarshad, Ghavami, S. Choobkar, and Wolfe, "Wireless sensor networks for monitoring physiological signals of multiple patients," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 5, no. 4, pp. 347–356, 2011 Mar 24.
- [26] Arsene, I. Dumitrache, and I. Mihu, "Expert system for medicine diagnosis using software agents," *Expert Systems with Applications*, vol. 42, no. 4, pp. 1825–1834, 2015 Mar 1.
- [27] T. Opsahl and P. Panzarasa, "Clustering in weighted networks," *Social Networks*, vol. 31, no. 2, pp. 155–163, 2009 May 1.
- [28] M. Hamim, S. Paul, S. I. Hoque, M. N. Rahman, and I. A. Baqee, "IoT based remote health monitoring system for patients and elderly people," in *Proceedings of the 2019 International Conference on Robotics, Electrical and Signal Processing Techniques, (ICREST)*, pp. 533–538, IEEE, Dhaka, Bangladesh, 10–12 January 2019.
- [29] F. A. Almalki and B. O. Soufiene, "EPPDA: An efficient and privacy-preserving data aggregation scheme with authentication and authorization for IoT-based healthcare applications," *Hindawi WCMC*, vol. 2021, Article ID 5594159, 18 pages, 2021.
- [30] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, pp. 13–16, Association for Computing Machinery, Helsinki Finland, 17 August 2012.
- [31] S. A. Haque, S. M. Aziz, and M. Rahman, "Review of cyber-physical system in healthcare," *International Journal of Distributed Sensor Networks*, vol. 10, no. 4, pp. 217–415, 2014 Apr 27.
- [32] B. Nathali Silva, M. Khan, and K. Han, "Big data analytics embedded smart city architecture for performance enhancement through real-time data processing and decision-making," *Wireless Communications and Mobile Computing*, 2017.
- [33] E. E. Egbogah and A. O. Fapojuwo, "A survey of system architecture requirements for health care-based wireless sensor networks," *Sensors*, vol. 11, no. 5, pp. 4875–4898, 2011 May.
- [34] P. Kumar and H. J. Lee, "Security issues in healthcare applications using wireless medical sensor networks: a survey," *Sensors*, vol. 12, no. 1, pp. 55–91, 2012 Jan.
- [35] S. Jabbar, F. Ullah, S. Khalid, M. Khan, and K. Han, "Semantic interoperability in heterogeneous IoT infrastructure for healthcare," *Wireless Communications and Mobile Computing*, vol. 2017, Article ID 9731806, 10 pages, 2017.
- [36] D. D. Koutsouris and A. A. Lazakidou, Eds., Springer, 2014 Sep 25, <https://link.springer.com/book/10.1007/978-3-319-06844-2>.
- [37] A. Datt, A. Goel, and S. C. Gupta, "Monitoring list for compute infrastructure in Eucalyptus cloud," in *Proceedings of the 2015 IEEE 24th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises*, pp. 69–71, IEEE, Larnaca, Cyprus, 15 June 2015.
- [38] A. Barrat, M. Barthelemy, R. Pastor-Satorras, and A. Vespignani, "The architecture of complex weighted networks," *Proceedings of the National Academy of Sciences*, vol. 101, no. 11, pp. 3747–3752, 2004 Mar 16.
- [39] M. Latapy, C. Magnien, and N. D. Vecchio, "Basic notions for the analysis of large two-mode networks," *Social Networks*, vol. 30, no. 1, pp. 31–48, 2008 Jan 1.
- [40] M. Kaiser, "Mean clustering coefficients: the role of isolated nodes and leaves on clustering measures for small-world networks," *New Journal of Physics*, vol. 10, no. 8, Article ID 083042, 2008 Aug 29.
- [41] Sacerdoti, Katz, Massie, and Culler, "Wide area cluster monitoring with Ganglia," in *Proceedings of the 2003 Proceedings IEEE International Conference on Cluster Computing*, pp. 289–298, IEEE, Hong Kong, China, 1 December 2003.
- [42] D. Barmpoutis and R. M. Murray, "Networks with the smallest average distance and the largest average clustering," <https://arxiv.org/abs/1007.4031>.
- [43] N. Deepa, B. Prabadevi, P. K. Maddikunta et al., "An AI-based intelligent system for healthcare analysis using Ridge-Adaline Stochastic Gradient Descent Classifier," *The Journal of Supercomputing*, vol. 77, no. 2, pp. 1998–2017, 2021.
- [44] C. Dhanamjayulu, U. N. Nizhal, P. K. R. Maddikunta et al., "Identification of malnutrition and prediction of BMI from facial images using real-time image processing and machine learning," *IET Image Processing*, 2021.
- [45] G. Thippa Reddy, A. Srivatsava, K. Lakshmana, R. Kaluri, S. Karnam, and G. Nagaraja, "Risk prediction to examine health status with real and synthetic datasets," *Biomedical and Pharmacology Journal*, vol. 10, no. 4, pp. 1897–1903, 2017.
- [46] L. Matthew, Massie, N. Brent, and b Chun, "the ganglia distributed monitoring system: design, implementation, and experience " Elsevier," *Parallel Computing*, vol. 30, pp. 817–840, 2004.