International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Optimization of Health-as-a-Service Using OptiFog Algorithm

Pratik Kanani[1]* **Anil Vasoya[2]** **Kamal Shah[2]** **Neel Kothari[1]** **Nilesh Patil[1]**
Gayatri Pandya[1] **Mamta Padole[3]**

[1]*Dwarkadas J. Sanghvi College of Engineering, Mumbai, India*
[2]*Thakur College of Engineering and Technology, Mumbai, India*
[3]*The Maharaja Sayajirao University of Baroda, Vadodara, India*
* Corresponding author's Email: pratikkanani123@gmail.com

Abstract: The healthcare industry relies on efficient and fast decision making. This paper aims to expand the Fog computing and Distributed computing domains to optimize quality of service (QoS) in order to facilitate IoT based healthcare applications with low latency requirements and developing a smart fog gateway equipped with an optimized fog algorithm. The purpose of this study is to optimize real-time healthcare data processing using Fog computing, ensuring dependable, rapid decision-making while minimizing delays caused by data transmission and computation. This is also known as Health-as-a-service (HaaS). We conduct an electrocardiography (ECG) analysis utilizing three computing paradigms: Cloud computing, Fog computing, and a heterogeneous distributed Fog computing setup employing the dynamic OptiFog algorithm. This algorithm effectively manages computational resources within the distributed Fog environment, utilizing Raspberry Pi clusters to enhance performance during worst-case scenarios. The response time is measured using Short Message Service (SMS). The OptiFog node exhibited a response better than the Fog node and the cloud node. The OptiFog algorithm not only takes into account different computing parameters like number of cores, memory usage, CPU utilization and response time of the computing node but also assigns dynamic priorities to these parameters to get the best possible processing available. Based on the workload of the task/node, it dynamically decides the job size to save the network bandwidth and to reduce the network overhead. In conclusion, the proposed work demonstrates that optimizing Fog computing with the dynamic OptiFog algorithm is an effective approach to meet low-latency requirements in IoT-based healthcare applications making it a valuable addition to the Health-as-a-Service (Haas) framework for real-time healthcare data processing.

Keywords: Cloud computing, Fog computing, OptiFog algorithm, Healthcare, Real-time ECG analysis.

1. Introduction

Healthcare as a Service" (HaaS) involves providing healthcare services through various paradigms. Like other "as a Service" models—such as Software as a Service (SaaS) or Platform as a Service (PaaS)—HaaS aims to offer healthcare-related solutions that are more adaptable, scalable, and budget-friendly. This approach capitalizes on networking advancements, making it possible to access, control, and provide different healthcare components remotely in a fault-tolerant and timely manner leveraging paradigms like Cloud, Fog, Edge, Distributed, Grid computing, P2P networks etc.

In recent times, there has been a significant expansion in the Internet of Things (IoT), leading to an unparalleled increase in the quantity and diversity of streaming data [1]. This interconnected network of objects with distinct identities facilitates advanced application services in various fields. In spite of its benefits, the IoT faces difficulties when it comes to handling extensive data quantities because of its restricted storage and processing capabilities.

To address these limitations, cloud computing has emerged as a promising solution, offering virtually limitless storage and processing resources on a pay-as-you-go basis [2]. By utilizing cloud technology, the IoT can offload its data-intensive tasks and take advantage of improved capabilities. But because of issues with latency, some apps and services have not completely embraced the advantages of cloud architecture.

The expansion of internet-connected smart devices and the growing number of services have placed considerable strain on network bandwidth, leading to a decline in QoS. The considerable delay in network connectivity between smart devices and the cloud makes it impractical to use delay-sensitive applications [3]. In response to these challenges, Fog computing [4] has emerged as a promising approach, extending cloud resources to the network edge [5]. Fog computing offers distributed services and facilitates data analysis and knowledge generation from streams produced by IoT devices. This potential has particularly valuable implications for pervasive healthcare monitoring applications [6], where IoT plays a critical role in continuously monitoring the physiological status of hospitalized patients without the need for constant caregiver involvement [7].

Wireless body area networks (WBAN) are a crucial technology in healthcare IoT, enabling seamless and unobtrusive acquisition of physiological information like electromyography (EMG), ECG, blood pressure, and blood temperature [8]. In order to effectively support pervasive healthcare applications, previous studies have explored cloud computing technology for IoT devices [9]. While conventional Fog computing approaches [10] have offered various solutions to address latency concerns in different application scenarios, their practical implementation in real-world pervasive healthcare computing is still in its early stages.

In this study, the researchers conduct a comparison of an ECG-based analysis system capable of real-time identification of abnormalities in ECG signals. The system employs Cloud, Fog, and Optimized Fog technology, based on the OptiFog algorithm [11] to carry out the analysis efficiently. The timely analysis of critical ECG data is of utmost importance, as it directly impacts the patient's life.

Initial cloud computing installation of the ECG-based healthcare system includes rigorous examination and recording of numerous QoS criteria. Subsequently, experiment is implemented on a basic Fog Computing node, and the QoS parameters are once again recorded. Finally, the OptiFog algorithm is employed, which significantly improves the computation performance using the Distributed Fog Computing environment. By comparing these three architectures, the researchers analyze the advantages of the OptiFog Algorithm, which includes reducing the burden on the cloud, conserving network bandwidth, enabling faster medical treatment through timely notifications like SMS and pull and push services.

Table 1. List of abbreviations

Abbreviation	**Full Form**
QoS	Quality of Service
IoT	Internet of Things
Haas	Health-as-a-Service
ECG	Electrocardiography
SMS	Short Message Service
SaaS	Software as a Service
PaaS	Platform as a Service
WBAN	Wireless Body Area Networks
EMG	Electromyography
BP	Blood Pressure
CPU	Central Processing Unit
SLAs	Service Level Agreements
GUI	Graphical User Interface
VPN	Virtual Private Network
WANs	Wide Area Networks

The Fog computation is performed by the Raspberry Pi model 3 B+. For optimized Fog computation, a heterogeneous fog computing system (using the OptiFog Algorithm) is utilized, and its detailed architecture is discussed later in section 3.4. The ECG signals are sent from a common source to the basic Fog, OptiFog and Cloud computing nodes. All three nodes conduct the same set of ECG interval calculations and analyze the signal for abnormalities. In case any abnormality is detected, the hospital and relatives are promptly notified through the SMS. The contributions of this research are:

1) To expand the Fog computing domain to optimize quality of service (QoS) in Fog computing in order to facilitate IoT based healthcare applications with low latency requirements and developing a smart fog gateway equipped with an optimized fog algorithm

2) To introduce an OptiFog algorithm centric computation node that is more efficient and has a lesser overall response time as compared to Fog and Cloud computation. This setup proves advantageous by conserving network bandwidth, preventing unnecessary cloud data uploads, eliminating transmission delays, reducing overall response time, and offering rapid and dependable computations.

The rest of the paper is organized as follows; the literature survey is mentioned in section 2. Section 3 talks about the proposed methodology and section 4 mentions the implementation procedure and experimental results obtained. Section 5 includes the conclusion.

2. Literature survey

The literature review studied targets systems in all three domains which are compared in the research: cloud computing, Fog computing and Distributed computing systems. It focuses on the latest research conducted in all of the mentioned systems individually.

The provision of healthcare is an area where data access, processing, analysis, sharing, and storage between various parties is exceedingly important. As a result, cloud computing has been utilized in the widest range of industries [12]. A conventional sensor-cloud architecture is a good strategy for handling low information rates and is used by the monitoring system to transfer measurements to the cloud for processing.

Goossens W et al. [13] employ an existing algorithm to predict blood pressure (BP) through real-time ECG monitoring in order to evaluate the advantages of edge computing. The ECG data is collected by a wearable monitoring node, wirelessly sent to an IoT cloud, and then stored on an SD card for offline use.

The design, development, and implementation of preliminary research about wireless ECG monitoring using an Android phone as a hub to connect it to the cloud server has been presented by M. I. Rizqyawan et al. [14]. A system is created that captures the user's ECG signal, sends it to the smartphone acting as a hub, saves it locally, uploads it to the server, and then displays it in the front-end Web application.

A cloud-based ECG monitoring and fibrillation detection for the healthcare system has been proposed by N. Gaigawali et al. [15]. Three distinct elements make up the overall system; the first part has an ECG acquisition sensor. The cloud platform, which is included in the second half, allows for the storage and analysis of data. The ECG smartphone application is included in the final section. The problem-solving methodology used in this study can increase cloud-based ECG monitoring's diagnostic precision, and cloud computing makes data analysis more manageable so that advancements in personalized healthcare can be made.

Fog computing's main goal, on the other hand, is to guarantee low and predictable latency in IoT applications that require it, such as healthcare services.

S. Malik et al. [16] made an effort to reduce network lag when gaining access to a centralized cloud environment. The main objective of this effort is to improve network performance by creating micro data centres for dispersed cloud resources. These micro data centres are strategically distributed based on network latency considerations to ensure an acceptable level of network performance. A new architecture to migrate some data centre tasks to the edge of the server has been introduced by Y. Shi et al. [17]. It furnishes rational intelligence to terminal devices while sifting through data for data centres. Regarded as an innovative structure, it offers constrained computing, storage, and networking functions in a distributed manner, spanning from end devices to the conventional cloud computing Data Centres.

Fog computing is a scalable solution to cloud computing that can store and process near-the-edge devices has been proposed by H. J. de Moura Costa et al. [18]. In this context, they present Fog computing as an alternative to reduce health data management complexity, consequently increasing its reliability. They classify the arrhythmias from ECG signals and process ECG signals too. An 'iFogSim' model that extends the cloud services to the edge of the network and decreases the network congestion by enhancing the resource management techniques, which perform real-time analytics and identify the optimal place of applications on the edge devices has been proposed by H. Gupta et al. [19]. The architecture of Smart Gateway with Fog Computing has been presented by M. Aazam et al. [20]. The architecture focuses on how data can be pre-processed and trimmed before sending to the cloud.

Shih-Nung Chen et al. [21] implement a complete distributed computing platform based on peer-to-peer file-sharing technology. The platform integrated scheduling, load balancing, file sharing, maintenance of data integrity, and user-friendly interface functions. G. L. Stavrinides and Helen D. Karatza [22] find a solution to the issues with scheduling simultaneous jobs on a cluster of distributed processors. The paper assesses the effectiveness of three distinct scheduling algorithms with two routing schemes.

Raspberry Pi (R-Pi) can be used as a fog computing node has been proposed by Bharathi P.D et al. [23]. To enhance the computational capacities of the Fog Computing layer, an option is to employ a collection of Raspberry Pi units functioning as a cluster. This setup consists of multiple Raspberry Pi devices, each with distinct configurations.

An algorithm that is used for scheduling in the cloud computing environment has been presented by Arash Ghorbannia et al. [24]. This method utilizes factors such as the current state of the processor to determine the appropriate node for job allocation. The objective is to establish a proficient scheduling technique that reduces the total processing duration of all tasks by evenly distributing them across the

Received: September 26, 2023. Revised: November 7, 2023.

Table 2. Future challenges/improvements in the literature review

Ref	**Technology used**	**Challenges/ Improvement in terms of**	**Solution given through the proposed methodology**
[2]	Integrating IoT and Cloud, using different cloud computing services	Pervasive Health care system generates Vast amount of data Needs Intelligence in IoT	Reducing data volume to save the network bandwidth and to ease the cloud burden
[4]	Discusses benefits of Fog computing in IoT	management in Fog node	Smart decision making due to dynamic parameter priority assignment.
[5]	Integrating edge and Cloud computing	Application development techniques are required, including those for distributed bandwidth management, failure recovery, and where to put application instances.	Clusters used thus making it more reliable. The job assigned to the Master Node is distributed among the Slave nodes, guaranteeing bandwidth management.
[6]	Health care fog node functions	Implementation of local storing and computing in fog Body area network (BAN) use of fog	BAN cannot account for future additions of data. Proposed methodology allows future data access and analytics by using the cloud technology.
[7]	IoT in Health care	incorporating real-time sensing data into medical records	Proposed methodology enables cloud based global storage and fog based local storage.
[9]	Cloud computing and IoT in general health care	The provision of healthcare via mobile devices has difficulties like: Data management and storage Data privacy and security universal access	Proposed methodology ensures service delivery through SMS (in absence of the internet) through a web-application portal (cloud based access) in presence of the internet.
[10]	Analysis and Survey of Fog Computing	Application-aware provisioning, reliability, networking, capacity, and security	Proposed methodology ensures context based ECG signal processing
[16]	Low latency support communication in cloud computing	Enable NAT and firewall communication. Automatically identifying and resolving connectivity issues	Proposed methodology uses secured protocols and encryption and decryption concepts.
[20]	Fog computing and IoT	providing more diverse services for QoS in fog computing Device mobility factor	Handheld devices like R-pi guarantee portability.
[19]	IoT and Fog computing	SLA-aware flow placements and resource scheduling based on simulation	OptiFog algorithm guarantees and takes into account SLA-level agreements.

accessible processors. Dankan Gowda et al. [25] propose an integrated hardware system with software programs to ensure QoS in healthcare systems using microcontroller which considers four parameters: CPU usage, response time, memory and number of cores utilized. Processing packages are designed to communicate through the IoT based on the accident reduction model (ARM) and augmented data recognition (ADR) system. Beagle cluster is proposed for minimised delay in computation. Mobility of devices, network bandwidth conservation, result optimization, data security, and computing power optimization of the fog node are all improved however the overall response time is an important aspect, which is considered in our methodology.

Table 2 lists the summary of the literature survey in terms of the technology used and possible improvements of the surveyed systems.

2.1 Research gap

1. Limitations in computational power of fog computing: Fog computing exhibits certain limitations in its computational power, which can impact its performance and efficiency.
2. Enhancing fog computing via fog clusters and distributed computing: A viable approach to enhance the capabilities of fog computing involves the implementation of a Fog cluster that utilizes distributed computing techniques to augment the overall computational capacity of Fog-based systems.
3. Selecting cluster type for computation within the Fog environment: Homogeneous or Heterogeneous: Both cluster types possess distinct advantages and disadvantages, prompting the need for an informed selection based on the specific requirements of the application.
4. Determining optimal job size for cluster nodes: static or dynamic: The determination of the optimal job size allocated to individual nodes within the cluster is a crucial aspect in achieving efficient Fog computing performance. This decision hinges upon whether to employ static or dynamic job sizes, necessitating an evaluation of the application's characteristics and workload dynamics to arrive at a well-suited solution.
5. Current approaches suffer from inefficiencies that result in delayed decision-making, impeding the timely delivery of medical care.

3. Material and methodology

This section is divided primarily into three parts: First explaining the process of decision-making then employing the OptiFog algorithm and lastly defining the architecture of the methodology.

3.1 Decision making

The ECG is a graphical representation of the heart muscles' electrical activity. This test is used to assess the heart's normal functioning, and any deviations from the normal rhythm are referred to as

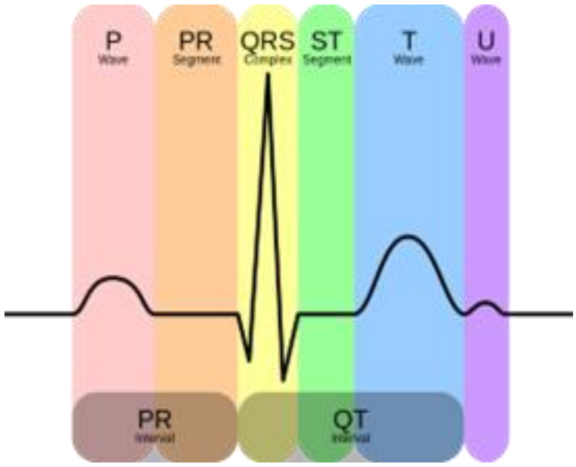

Figure. 1 An ECG wave in time domain

Table 3. Standard ECG intervals for a healthy adult with standard bpm of 60

Intervals	Normal Value	Normal Variation
QT Intervals	400 ms	±40ms
QRS Interval	100ms	±20ms
PR Interval	160 ms	±40ms

"Arrhythmia". ECG plays a crucial role in identifying irregular heartbeats, evaluating blood flow blockages caused by cholesterol buildup, and detecting heart enlargements [26]. The primary intervals in the ECG waveform are PR and QT. The PR wave corresponds to the electrical impulse transmission from the right atrium to the left atrium. The QRS complex appears when both ventricles start pumping, typically accompanied by a "beep" sound on cardiac monitors. The ST segment follows the initial contraction, and the T wave is generated during ventricular relaxation.

The normal beats are from 60-100 with these intervals present in the ECG waveform [27]. These intervals, PQRST [28], are shown in Fig. 1.

After analyzing various time intervals and cross-referencing the data with Table 3, the presence of normal or abnormal waves can be identified [30] i.e., if the value does not fall within the mentioned range, it is termed as abnormal. The algorithm presented in the following text represents the computational duty assigned to each individual node for every wave within the designated set. In the event of detecting an abnormal wave among the assigned set, a signal is activated across all three computing nodes i.e., cloud, fog, and OptiFog node.

The Windowing algorithm is chosen since it is an array-based approach, and it takes less time for computation with 99% accuracy [29]. Also, since the signal is a time-series, there is no need for conversion from time to the frequency domain, which could increase the time.

Received: September 26, 2023. Revised: November 7, 2023.

Algorithm 1 Windowing algorithm

```
{
// Input: ECG Wave in time domain
// Output: Different time intervals of ECG wave
// i -> represents each ECG wave
// n -> total no. of ECG waves
// P, Q, R, S, T: Reference points of ECG waves
// P-R, R-R, QRS, Q-T: Different time intervals
of an ECG wave
// fs: Sampling frequency in Hz
// bpm: beats per minute
i <- 1
while(i<=n)
{
trr = R-to-R interval = |(R[i]-R[I+1])/fs|
tpr = P-to-R interval = |(P-to-R)/fs|
tqrs = QRS interval = |[(Q[i]-8)-(S[i]-8)]/fs|
tqt = Q-to-T interval = [T[i]+(trr*0.13)-(Q[i]-8)]/fs
bpm = (trr*60)/fs
  }
//end while
}
```

3.2 OptiFog algorithm

The OptiFog Node employs the OptiFog algorithm [11] for scheduling tasks in a diverse Fog computing setting. The main objective is to achieve optimal performance in heterogeneous environment by effectively utilizing the available processing power within the system.

3.2.1. Insights to the OptiFog algorithm

The The OptiFog algorithm is a unique and effective optimization technique that employs four distinct techniques, namely memory-based, response-time-based, CPU-usage-based, and number-of-cores-based, to analyze the performance of the scheduling algorithm. Each technique was executed when nodes are engaged in other computational tasks, allowing for a comparison of their respective weights in heterogeneous computing scenarios. The computation health status factor (impact factor) (ψ) thus calculated is assigned to each node in the distributed system, and it determines the allocation of jobs to each node simultaneously. During the process, every node sends information about its current CPU usage, cores, and memory status to the master node. Afterward, the primary node calculates response time and impact factor values for each node in every cycle. It uses the impact factor value to decide the number of jobs to assign.

The CPU and core capacities of each node vary, depending on factors such as operating frequency, processor specifications, cache size, and bus size. These specifications reflect the processing capabilities of a node, making it a crucial factor in determining priority (₽) for job allocation. On the other hand, the memory and response time of a node is measured in units like GB and ms, allowing for comparisons with other nodes in the distributed system. Consequently, these collective units of memory and response time play a significant role in the overall optimization process.

Impact factor pertains to a node's comprehensive ability encompassing memory, CPU, cores, and response time. However, OptiFog employs three primary criteria to determine this impact factor. That is Capacity Factor (Ͼ), Memory (μ) and Time (Ł).

a) Capacity Factor (Ͼ): This metric relies on the method of assessing CPU usage and the count of inactive processing cores. The level of CPU usage is interconnected with the core count. Both of these approaches contribute to a notably strong performance, exhibiting nearly identical outcomes. Consequently, in this scenario, these two measures are combined to compute the resulting factor as:

$$\text{Ͼ} = NC * (1 - \mathrm{CU}), \text{where } \mathrm{CU} \in [0,1] \quad (1)$$

Where CU = core usage and NC = Average of available cores on the node

b) Memory Factor (μ): In this aspect, when is smaller than, the performance of the node improves. As a result, each node is identified and normalized to a value of 1. The calculation for this factor is as follows:

$$\mu = \frac{(1-MU_i)}{\sum_{i=1}^{n} MU_i} \quad (2)$$

Where MU= Memory usage

c) Time Factor (Ł): The relationship between a node's response time and its capacity is such that the response time decreases as the capacity increases. Bearing this principle in mind, the ranking factor is formulated to assign lower ranks to nodes with higher response times and higher ranks to nodes with lower response times.

$$\text{Ł} = \frac{\sum_{i=1}^{n} RT_i}{RT_i} \quad (3)$$

Where RT=Response time

After finding Ͼ, μ and Ł. The final ψ is calculated as:

 DOI: 10.22266/ijies2024.0229.36

Received: September 26, 2023. Revised: November 7, 2023.

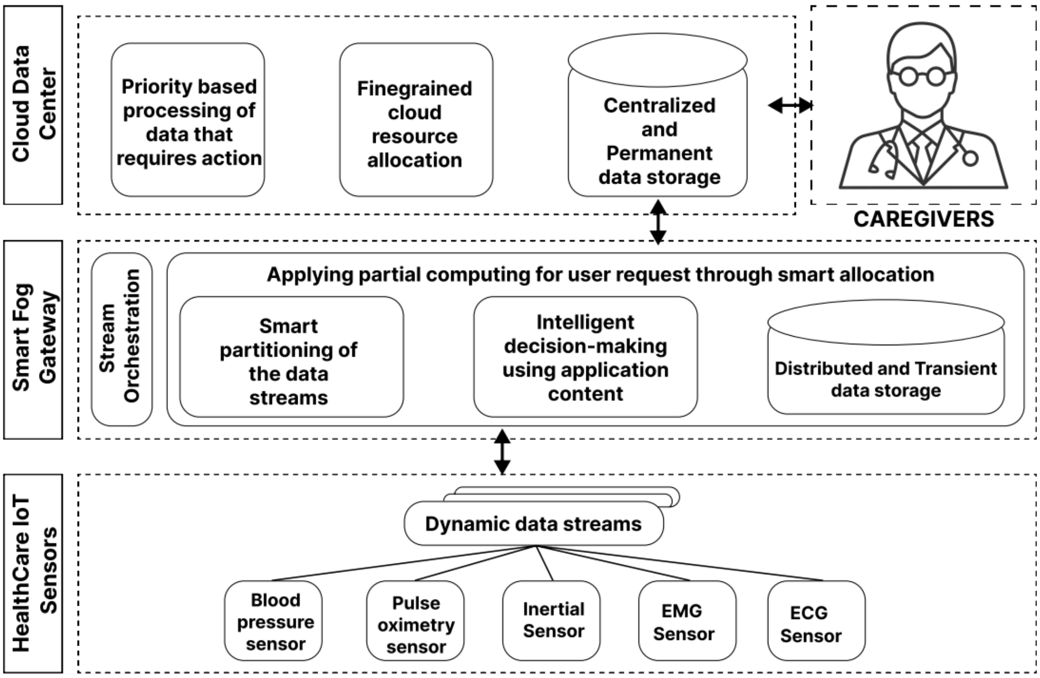

Figure. 2 A smart fog computing gateway

$$\psi = 3Ͼ + 2Ł + 1\mu \qquad (4)$$

where, numericals ϵ {Ᵽ}

The OptiFog Algorithm essentially makes sure that an optimal batch-size is taken during each process in order to get a good performance from the Dispy system. It considers four primary parameters: CPU usage, Response time, Memory usage, Number of cores. The same job is given to nodes and each of the parameters are assessed. Based on the observations, the impact factor ψ is determined. At any given time, the OptiFog algorithm dynamically assigns jobs based on the workload of each operating node thus ensuring bandwidth management. It is capable of assigning an optimal job-size for each node based on the situation. It has been designed considering worst-case scenarios and thus has a good performance in a normal case.

3.3 Smart fog computing architecture

In recent times, rapid advancements in pervasive and context-aware computing have led to the integration of the IoT into our daily lives. The IoT refers to a network of smart devices equipped with software and hardware offering diverse and advanced services ubiquitously. This integration has brought about the need for improved infrastructure and sophisticated mechanisms for service discovery, resource management, and energy management. Energy-constrained IoTs, operating independently, face various challenges in dealing with the continuous generation of data streams. To overcome these challenges, the integration of cloud computing with IoTs has proven crucial. However, due to the high network latency of centralized cloud servers, this integration poses its own unique set of challenges, particularly for delay-sensitive applications like healthcare. To address this issue, smart gateways like Fog computing are essential, as they extend cloud services to the network's edge, minimizing delays. By leveraging application context and service level agreements (SLAs), this approach ensures QoS. Fig. 2 illustrates the proposed methodology which essentially emulates “data-on-the-go”.

The proposed methodology involves partial computing of user requests through smart allocation in a basic Fog gateway. The optimal placement of applications on Fog resources is determined through careful analysis. In the realm of cloud computing, the approach involves allocating portions of computational information to cloud assets based on decision criteria that consider factors such as response speed, resource availability, and resource consumption. Furthermore, it prioritizes specific tasks within the requests that emerge from the distributed Fog environment

Ultimately, the proposed approach utilizes the OptiFog algorithm in the distributed Fog computing framework to minimize computation delay and ensure efficient performance, which is especially critical for life-saving applications like healthcare.

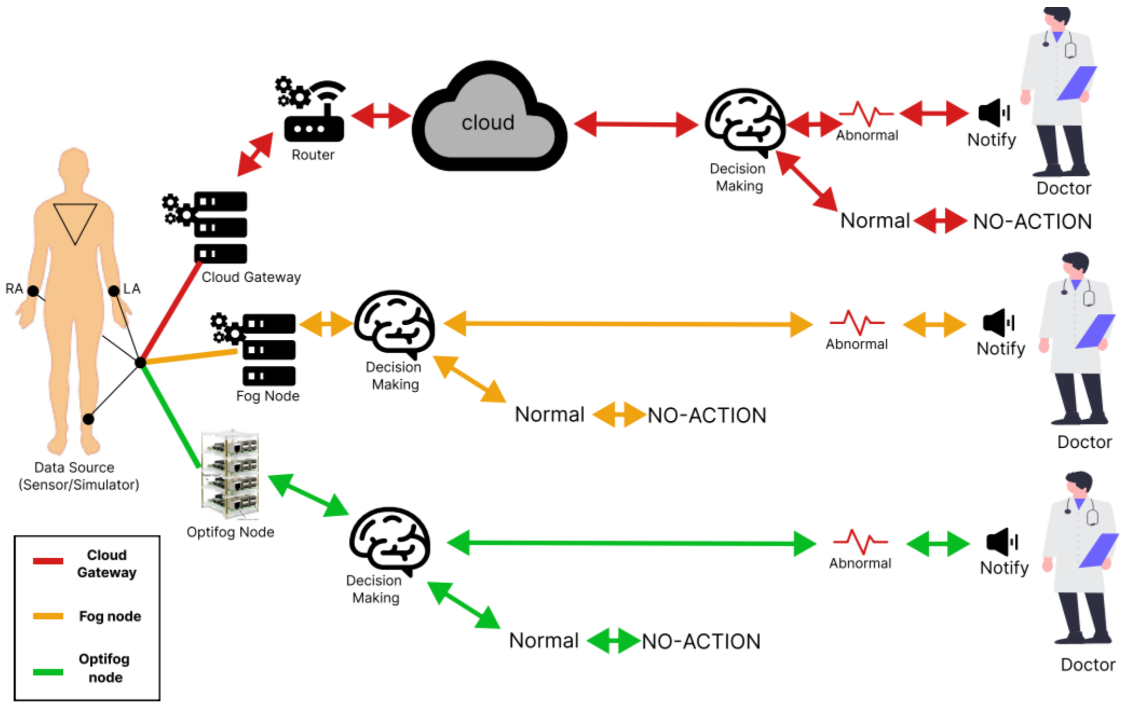

Figure. 3 Proposed system architecture

3.3.1 System architecture

The suggested method considers a real-time healthcare system that has a number of components, including a data source, fog node, OptiFog node, gateway, decision-making process, and messaging service. The real-time ECG signal is obtained from the patient. These signals are then transmitted to the Cloud and both the Fog Computing nodes for recording and analysis. To identify the reference points PQRST in the ECG signal, the windowing algorithm [29] is employed. Subsequently, ECG time intervals are determined based on these reference points. The next step involves decision making to determine whether the given ECG signal is normal or abnormal.

In the suggested design, as illustrated in Fig. 3, the live ECG signals are sent concurrently to the Cloud, Fog, and OptiFog nodes. Each of these systems conducts an analysis on the identical signal, and a comparison of outcomes is made, considering factors such as processing duration and transmission speed. If an anomaly is identified in the signal, a text message is promptly dispatched to the physician, including the time when it occurred, signal characteristics within the interval, and patient details. The effectiveness of response is subsequently evaluated using the provided timestamp.

4. Implementation and results

4.1 Configuration of the nodes

A total of 6 ECG waves are used in each sub-job for the Cloud and basic Fog node while the OptiFog node uses a variable job size as per the ψ determined by the OptiFog Algorithm.

4.1.1. Cloud node

An Amazon EC 2 instance is used for the Cloud node.

4.1.2. Basic fog node

Raspberry Pi Model 3 B+ is used and in order to make the basic fog computing node.

4.1.3. OptiFog node

The OptiFog Node, which is a cluster, is a master-slave architecture that comprises 3 nodes which are Raspberry Pi 3 B+ model and 1 node which is Raspberry Pi 4 model having 4 GB RAM. To implement the distributed computing for the Raspberry Pi cluster in fog computing, “dispy” is selected. Dispy is developed in Python and is used since Python works very well with Raspbian operating systems and makes a distributed system more scalable [11]. The architectural setup of the OptiFog node is given in Fig. 4 below.

Received: September 26, 2023. Revised: November 7, 2023.

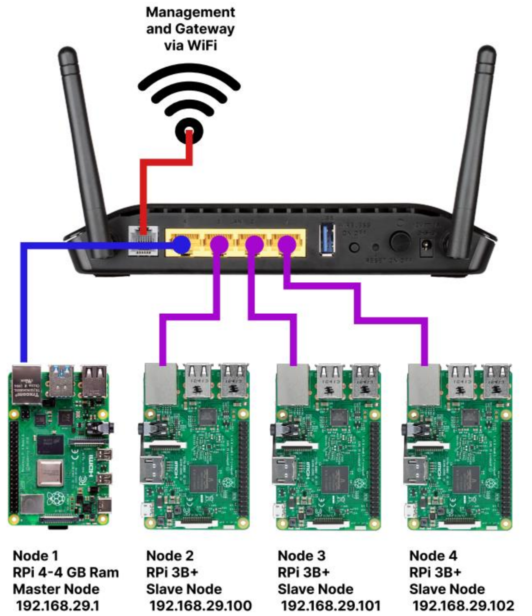

Figure. 4 Architectural setup of the OptiFog node

4.2 Defining different time stamps and defining the QoS parameters

QoS parameters like Transmission delay and Computation delay are compared and are as follows.

1) Transmission delay:

Network transmission delay encompasses the overall duration required for data to be transmitted between an initial location (source) and a final location (destination).

2) Computation delay:

The computation delay refers to the overall duration required for computation. In the suggested framework, it denotes the time elapsed from when the complete signal is received within the system until the processing is completed and the output, either normal or abnormal, is determined. By considering the computation time, it becomes possible to calculate the speed-up in Fog Computing.

4.3 Responsive interface

An interactive graphical user interface (GUI), depicted in Fig. 5, has been developed to delve deeper into the processing and analysis of ECG signals. This user-friendly web interface operates seamlessly on all the three nodes. The GUI displays essential information, including the patient's name, an indication of signal abnormality highlighted in red, the filename (with a ".txt" extension) storing the ECG signals for processing, various QoS parameters, the ECG waveform, patient particulars, and distinct intervals for each ECG wave, along with their corresponding abnormalities.

In the event of an abnormal signal detection, the system is programmed to automatically send an SMS notification to the designated Health care supporting staff, ensuring timely attention to the patient's condition. The web interface is implemented on both the basic fog computing Node and the OptiFog node also. However, there is a notable distinction: the former operates on the localhost at the Gateway. The ultimate outcomes of these three systems are transmitted via SMS messages, each containing crucial information, including the computing node of origin, patient name, mobile number, timestamp of message generation, the filename storing the signals, and the corresponding ECG intervals. This valuable data aids doctors and hospital staff in making advanced preparations before the patient's arrival. Please be aware that delays in network transmission by the service provider have the potential to influence the arrival time of the SMS, leading to possible response time delays.

When examining the millisecond timestamps depicted in Fig. 6 and subsequently transformed according to [31], they correspond to distinct timings: 1692510780787 for Fog, 1692510780212 for OptiFog, and 1692510787732 for cloud. This observation highlights the noteworthy capability of the OptiFog node, which exhibits a response lead of approximately 575 milliseconds over the Fog node and 7520 milliseconds over the cloud node. This time advantage holds crucial implications, particularly in critical patient care scenarios.

The experimentation involved a singular patient scenario in the Fog, OptiFog, and cloud computational framework. The resulting parameters and data are comprehensively detailed in Table 6.

The results indicate that OptiFog computing outperforms both basic Fog and Cloud computing in terms of overall response time performance, demonstrating superior response time performance.

The system's functionality is assessed through experiments involving different patient quantities, enabling a thorough exploration of its behaviour.

This involves considering and presenting the average parameter values for a group of 'n' patients, fostering subsequent discussion.

4.4 Transmission delay

The time it takes for data to travel from the source to the destination, known as the Transmission Delay, is influenced by various factors. Some of these factors are:

i) Number of intermediate stops (Hops): This factor refers to the number of devices or network nodes a data packet must traverse between its source

Received: September 26, 2023. Revised: November 7, 2023.

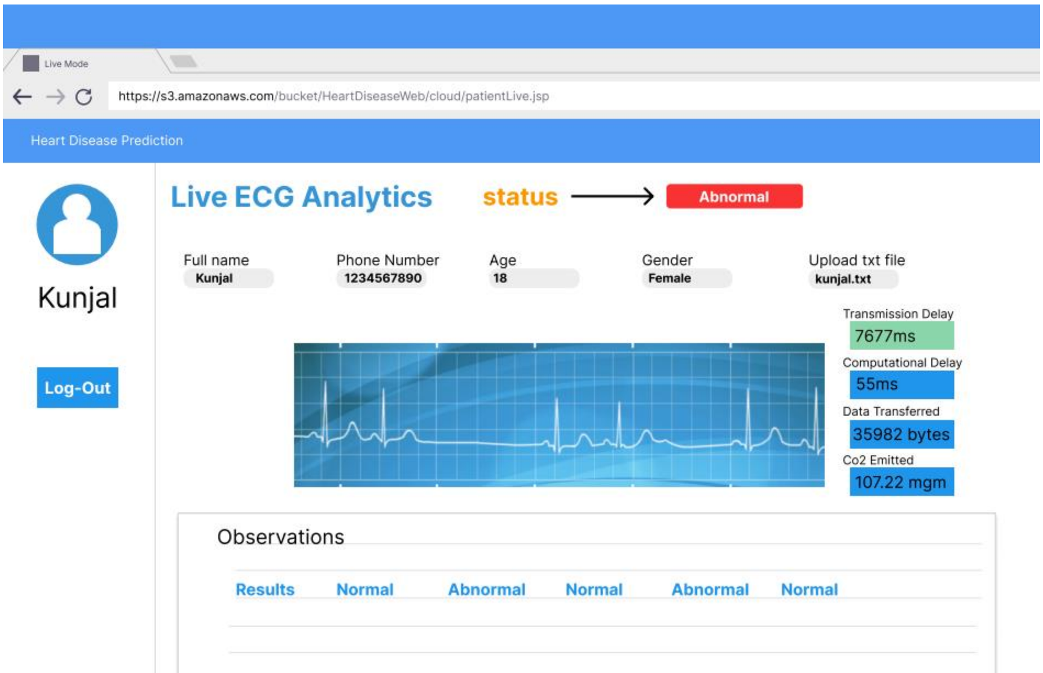

Figure. 5 System interface for analyzing ECG signal on cloud computing

Table 6. QoS parameters and their values for a single patient's data analysis

Parameters	**Cloud Computing**	**Fog Computing**	**OptiFog Computing**
Transmission Delay (ms)	7677	117	**117**
Computational Delay (ms)	55	670	**95**
Response Time in ms (Time Format in Java)	1692510787732	1692510780787	**1692510780212**

and destination. Each hop adds a slight delay as the packet moves through the network.

ii) Capacity of the network's available bandwidth: The available network bandwidth determines how much data can be transmitted simultaneously. Limited bandwidth can lead to congestion and increased transmission delay.

iii) Layer conversions: When data passes through different network layers, such as from the application layer to the physical layer, conversions may be necessary. These conversions can introduce delays in data transmission.

iv) Virtual private network (VPN) setup: VPNs provide secure communication over public networks, but the encryption and decryption processes involved can introduce additional transmission delay.

v) Configurations for wired and wireless connections: The choice between wired and wireless connections affects transmission delay. Wireless connections often have higher latency compared to wired connections.

vi) Instances of network congestion: Network congestion occurs when there is more data traffic than the network can handle efficiently. During congestion, data packets may be delayed as they wait for their turn to be transmitted.

vii) Tunnelling: Tunnelling involves encapsulating data packets within other packets for secure transmission. This process can introduce overhead and delay in data transfer.

viii)Number of active users: The number of users actively utilizing the network can impact transmission delay. As more users share the network, resources become limited, potentially leading to longer transmission times for data packets.

Each hop corresponds to a point where a data packet moves from one network segment to another. During its journey between the source and destination, data packets pass through routers [31]. Wide Area Networks (WANs), which typically cover vast geographic regions and connect numerous devices, may experience a slight reduction in network bandwidth due to increased delivery time for packets from source to destination. Consequently, with the increase in number of hops, there is a higher likelihood of experiencing decreased bandwidth capacity. In both fog and OptiFog computing nodes, the number of hops remains constant at one. However, Cloud computing exhibits variability in the number of hops due to its reliance on the WAN.

Received: September 26, 2023. Revised: November 7, 2023.

(a) (b) (c)

Figure 6. SMS responses for: (a) Cloud, (b) Fog, and (c) OptiFog nodes

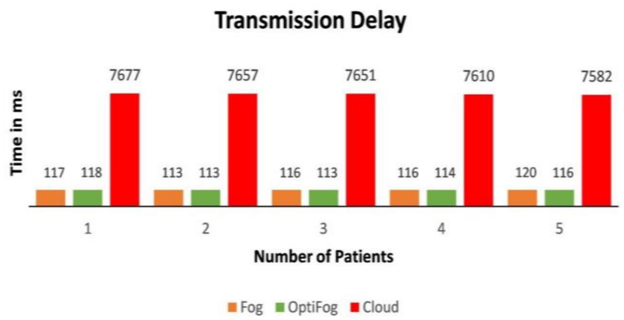

Figure. 7 Transmission delays for fog, OptiFog and cloud nodes

The disparity between Cloud and Fog computing in terms of transmission delay is clearly discernible, as illustrated in Fig. 7. The transmission delay of the basic Fog node is however comparable to the OptiFog node and it is observed that the value remains almost the same for multiple patients.

4.5 Computational delay

The performance potential of a device is contingent upon its hardware arrangement, which includes elements like:

i) Cache Storage: Cache storage is a high-speed, small-sized memory storage unit within a device that stores frequently accessed data to expedite data retrieval, enhancing overall performance.

ii) Processor model: The processor model refers to the specific type and model of the central processing unit (CPU) within a device. Different processor models offer varying levels of processing power and capabilities.

iii) Clock speed: Clock speed measures how quickly a CPU can execute instructions, typically measured in gigahertz (GHz). A higher clock speed indicates faster processing, which can lead to improved device performance.

iv) Task scheduling method: Task scheduling methods dictate how a device manages and prioritizes tasks or processes. Effective scheduling methods can optimize resource allocation and enhance multitasking capabilities.

v) Data transmission pathway: The data transmission pathway refers to the route and technology used for transferring data within a device or between devices. Efficient pathways can minimize latency and improve data transfer speeds.

vi) Memory size: Memory size indicates the amount of random-access memory (RAM) available in a device. A larger memory size allows for more simultaneous data storage and retrieval, positively impacting performance.

vii) Core count: Core count refers to the number of processing cores within a CPU. Devices with multiple cores can execute multiple tasks in parallel, leading to enhanced multitasking and overall performance.

In the present context, the basic fog computation is performed by the fog device, which can efficiently read and analyze multiple real-time ECG waves simultaneously across different ports. On the other hand, the OptiFog node undertakes similar tasks within a distributed environment, employing the OptiFog algorithm. To assess its performance, the Fog node's average computational delay is evaluated while varying the number of patients being processed, as depicted in Fig. 8.

The delay in computation is nearly the same when considering cloud computing, but it varies significantly with Fog computing. In fog computing, the growth follows a polynomial pattern, specifically

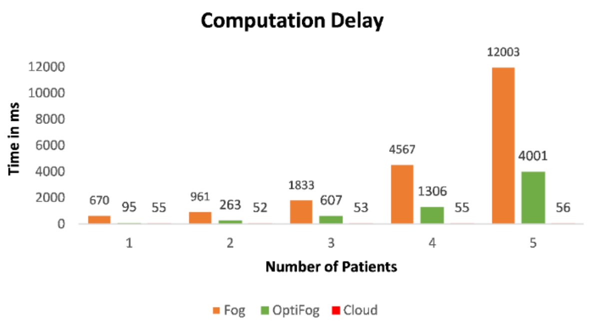

Figure. 8 Computation delays for fog, OptiFog and cloud nodes

of order 3˚. The computational delay is also influenced by various factors, such as GUI computation, background processes, refresh rate, and the number of concurrent tasks. For a group of four patients, the individual computational delays are 4524, 4661, 4486, and 4594 milliseconds, their average is calculated as 4567 milliseconds. Similarly, when considering the OptiFog node, the same averaging is done for computation delay values, for the same group of patients. The average computation delay of the four patients is 1306 milliseconds, as also illustrated in Fig. 8.

Consequently, while overloading the Fog system isn't advisable for time-sensitive decision-making setups, the use of an OptiFog node remains a viable option.

5. Conclusion

The real-time processing of ECG signals has implications beyond merely discarding unnecessary data on the cloud. It introduces transmission delays that could lead to medical emergencies and hinder timely patient treatments. While fog computing addresses these delays and offers quicker responses compared to cloud-based solutions, it falls short in terms of computational power. Enhancement of Fog computing can be achieved through the use of Fog clusters. However, these clusters perform optimally when implementing heterogeneous fog computing alongside the OptiFog algorithm.

The OptiFog algorithm is specifically crafted to assess the computational capabilities of individual nodes within a cluster. Using this health parameter as a basis, tasks are allocated with sizes proportionate to each node's health factor. This dynamic algorithm continually monitors node health during task execution, adapting job sizes accordingly. To determine node health, factors such as CPU usage, core count, response time, and available memory are taken into account in the same order of priority. The OptiFog algorithm ensures a minimum improvement of 14% in computations in worst case scenarios when the system is under load.

In order to make an optimal decision, firstly the number of waves in the job are decided based on the frame size. This in turn saves the network bandwidth and reduces the number of job transactions. Moreover, the system constantly checks the number of jobs which increases or decreases as per the change in the impact factor. The response time is measured in milliseconds using short message service (SMS) and after experimentation it is observed that, though the computation delay of the node utilizing the OptiFog algorithm is greater than the cloud node, the OptiFog setup is more efficient and has a lesser overall response time. The OptiFog node exhibited a response lead of approximately 575 milliseconds over the fog node and 7520 milliseconds over the cloud node. Despite a higher computation delay compared to the cloud node, the OptiFog setup proved more efficient, achieving a lesser overall response time.

When assessing the performance of different configurations—cloud processing, Fog processing, and Fog cluster processing with the OptiFog algorithm—on varying quantities of ECG waves from diverse patients, the third configuration utilizing the OptiFog algorithm yields the most favourable outcomes. This setup proves advantageous by conserving network bandwidth, preventing unnecessary cloud data uploads, eliminating transmission delays, offering rapid and dependable computations and ensuring less power consumption. Furthermore, these systems are cost-effective, utilizing available system-on-chip (SoC) components, and leveraging clustering to enhance computation speed and reliability. Their portability allows deployment in diverse locations, even operating on battery power in remote areas.

Additionally, these systems possess the capability to store crucial data for historical analysis and sharing during instances of offline periods. Once back online, the data can be uploaded, stored, and shared on designated cloud storage platforms.

6. Future Work

Future work includes improving the computation delay of the OptiFog computing node to surpass that of Cloud computation. The current situation has the Cloud computation node having the least Computation delay as compared to the OptiFog and Fog computing nodes, thus leaving scope of improvement.

Improving the performance of the OptiFog computing node in the ECG analysis system can help

Received: September 26, 2023. Revised: November 7, 2023.

reduce response times and enhance the overall efficiency of the healthcare application. The following parameters and strategies can be considered to optimize the OptiFog computing node:
1. Resource allocation (CPU, memory, storage)
2. Caching: Implement data caching mechanisms to store and retrieve frequently accessed data. Caching can reduce the need for repetitive computations and database queries, resulting in faster response times.
3. Data compression: Compress ECG data before transmission to the cloud node and decompress it on arrival. This reduces the amount of data that needs to be transferred over the network, reducing latency.
4. Content delivery network (CDN): Consider using a CDN to cache and deliver static content (e.g., images, scripts) closer to end-users. This can improve the delivery of graphical elements in your application.
5. Network optimization: Optimize network configurations and use content delivery networks (CDNs) to reduce latency in data transmission between the cloud node and end-users.
By focussing on these parameters and strategies, the performance of the OptiFog computing node in the ECG analysis system can be enhanced, resulting in even faster response times and improved service delivery.

Conflicts of interest

The authors declare no conflict of interest.

Author contributions

All authors contributed to the study conception and design. Material preparation and data collection analysis were carried by Pratik Kanani, Neel Kothari, Kamal Shah, Anil Vasoya, Nilesh Patil, Gayatri Pandya, Mamta Padole. All authors read and approved the final manuscript.

References

[1] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey", *Computer Networks*, Vol. 54, No. 15, pp. 2787-2805, 2010.

[2] A. Botta, W. D. Donato, V. Persico, and A. Pescapé, "Integration of cloud computing and internet of things: A survey", *Future Generation Computer Systems*, Vol. 56, pp. 684-700, 2016.

[3] M. Hassanalieragh, A. Page, T. Soyata, G. Sharma, M. Aktas, G. Mateos, B. Kantarci, and S. Andreescu, "Health Monitoring and Management Using Internet-of-Things (IoT) Sensing with Cloud-Based Processing: Opportunities and Challenges", In: *Proc. of 2015 IEEE International Conference on Services Computing*, New York, NY, USA, pp. 285-2, 2015.

[4] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things", In: *Proc. of the First Edition of The Mcc Workshop on Mobile Cloud Computing*, pp. 13-16, 2012, [Online]. Available: https://doi.org/10.1145/2342509.2342513.

[5] H. Chang, A. Hari, S. Mukherjee, and T. V. Lakshman, "Bringing the cloud to the edge", In: *Proc. of 2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp.346-351, 2014, [Online]. Available: https://doi.org/10.1109/infcomw.2014.6849256.

[6] Y. Shi, G. Ding, H. Wang, H. E. Roman, and S. Lu, "The fog computing service for healthcare", In: *Proc. of 2015 2nd International Symposium on Future Information and Communication Technologies for Ubiquitous HealthCare (Ubi-HealthTech)*, pp. 1-5, 2015, [Online]. Available: https://doi.org/10.1109/ubi-healthtech.2015.7203325.

[7] S. M. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, and K. S. Kwak, "The Internet of Things for Health Care: A Comprehensive Survey", *IEEE Access*, Vol. 3, pp. 678-708, 2015, doi: 10.1109/ACCESS.2015.2437951.

[8] N. Bui and M. Zorzi, "Health Care Applications", In: *Proc. of the 4th International Symposium on Applied Sciences in Biomedical and Communication Technologies*, pp. 1-5, 2011. [Online]. Available: https://doi.org/10.1145/2093698.2093829.

[9] C. Doukas and I. Maglogiannis, "Bringing IoT and Cloud Computing towards Pervasive Healthcare", In: *Proc. of 2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, Palermo, Italy, pp. 922-926, 2012, doi: 10.1109/IMIS.2012.26.

[10] S. Yi, C. Li, and Q. Li, "A survey of Fog Computing", In: *Proc. of the 2015 Workshop on Mobile Big Data*, pp. 37-42, 2015, [Online]. Available: https://doi.org/10.1145/2757384.2757397.

[11] P. Kanani and M. Padole, "Optifog: Optimization of Heterogeneous Fog Computing for QOS in Health Care", *Journal of Theoretical and Applied Information Technology*, Vol. 98, No.22, pp. 3625-3642, 2020.

[12] D. Morais, F. G. Pinto, I. M. Pires, N. M. Garcia, and A. J. Gouveia, "The influence of cloud computing on the healthcare industry: a review of applications, opportunities, and challenges for

the CIO", *Procedia Computer Science*, pp. 714-720, 2020, [Online]. Available: https://doi.org/10.1016/j.procs.2022.07.106.

[13] W. Goossens, D. Mustefa, D. Scholle, H. Fotouhi, and J. Denil, "Evaluating Edge Computing and Compression for Remote Cuff-Less Blood Pressure Monitoring. Journal of Sensor and Actuator Networks", *Journal of Sensor and Actuator Networks*, Vol. 12, No. 1, pp. 1-13, 2023, [Online]. Available: https://doi.org/10.3390/jsan12010002.

[14] M. I. Rizqyawan, M. F. Amri, R. P. Pratama, and A. Turnip, "Design and development of Android-based cloud ECG monitoring system", In: *Proc. of 2016 3rd International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)*, Semarang, Indonesia, pp. 1-5, 2016, doi: 10.1109/ICITACEE.2016.7892444.

[15] N. Gaigawali and U. M. Chaskar, "Cloud Based ECG Monitoring and Fibrillation Detection for Healthcare System", In: *Proc. of 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, Madurai, India, pp. 287-291, 2018, doi: 10.1109/ICCONS.2018.8662999.

[16] S. Malik, F. Huet, and D. Caromel, "Latency based group discovery algorithm for network aware cloud scheduling", *Future Generation Computer Systems*, Vol. 31, pp. 28-39, 2014.

[17] Y. Shi, G. Ding, H. Wang, H. E. Roman, and S. Lu, "The fog computing service for healthcare", In: *Proc. of 2015 2nd International Symposium on Future Information and Communication Technologies for Ubiquitous HealthCare (Ubi-HealthTech)*, Beijing, China, pp. 1-5, 2015, doi: 10.1109/Ubi-HealthTech.2015.7203325.

[18] H. J. D. M. Costa, C. A. D. Costa, R. D. R. Righi, and R.S. Antunes, "Fog computing in health: A systematic literature review", *Health and Technology*, Vol. 10, pp. 1025-1044, 2020, [Online]. Available: https://doi.org/10.1007/s12553-020-00431-8.

[19] H. Gupta, A. V. Dastjerdi, S. K. Ghosh, and R. Buyya, "iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments", *Software: Practice and Experience*, Vol. 47, No. 9, pp. 1275-1296, 2017, [Online]. Available: https://doi.org/10.1002/spe.2509.

[20] M. Aazam and E. N. Huh, "Fog Computing Micro Datacenter Based Dynamic Resource Estimation and Pricing Model for IoT", In: *Proc. of 2015 IEEE 29th International Conference on Advanced Information Networking and Applications, Gwangju, Korea (South)*, pp. 687-694, 2-15, doi: 10.1109/AINA.2015.254.

[21] S. N. Chen, J. J. P. Tsai, C. W. Huang, R. Ming Chen, and R. C. K. Lin, "Using distributed computing platform to solve high computing and data processing problems in bioinformatics", In: *Proc. of Fourth IEEE Symposium on Bioinformatics and Bioengineering*, Taichung, Taiwan, pp. 142-148, 2004, doi: 10.1109/BIBE.2004.1317336.

[22] G. L. Stavrinides and H. D. Karatza, "Task Group Scheduling in Distributed Systems", In: *Proc. of 2018 International Conference on Computer, Information and Telecommunication Systems (CITS)*, Alsace, Colmar, France, pp. 1-5, 2018, doi: 10.1109/CITS.2018.8440181.

[23] P. D. Bharathi, V. Ananthanarayanan, and P. B. Sivakumar, "Fog computing-based environmental monitoring using nordic thingy: 52 and raspberry Pi", In: *Smart Systems and IoT: Innovations in Computing: Proceeding of SSIC 2019*, pp. 269-279, 2020. [Online]. Available: https://doi.org/10.1007/978-981-13-8406-6_27.

[24] A. G. Delavar, M. Javanmard, M. B. Shabestari and M. K. Talebi, "RSDC (reliable scheduling distributed in cloud computing)", *International Journal of Computer Science, Engineering and Applications*, Vol. 2, No. 3, pp. 1–16, 2012, [Online]. Available: https://doi.org/10.5121/ijcsea.2012.2301.

[25] D. Gowda V, A. Sharma, B. K. Rao, R. Shankar, P. Sarma, A. Chaturvedi, and N. Hussain, "Industrial quality healthcare services using Internet of Things and fog computing approach", *Measurement: Sensors*, Vol. 24, pp. 1-16. 2022. https://doi.org/10.1016/j.measen.2022.100517

[26] "Electrocardiogram (ECG)", *Blood Pressure Uk* [Online]. Available: http://www.bloodpressureuk.org/BloodPressureandyou/Medicaltests/ECG.

[27] "Standard range of intervals", *E Medicine* [Online]. Available: http://emedicine.medscape.com/article/2172196-overview.

[28] "PQRST INTERVAL", *Spark Fun* [Online]. Available:https://cdn.sparkfun.com/assets/learn_tutorials/2/5/0/EKG_Complex_en.svg.png.

[29] M. Umer, B. A. Bhatti, M. H. Tariq, M. Z. U. Hassan, M. Y. Khan, and T. Zaidi, "Electrocardiogram feature extraction and pattern recognition using a novel windowing algorithm", *Advances in Bioscience and Biotechnology*, Vol. 5, No. 11, pp. 886-894, 2014.

[30] P. Kanani and M. Padole, “Recognizing Real Time ECG Anomalies Using Arduino, AD8232 and Java”, *Advances in Computing and Data Sciences*, India pp. 54-64 2018, doi: 10.1007/978-981-13-1810-8_6.
[31] Available: https://currentmillis.com (Accessed: 24 August 2023).
[32] D. Comer and D. L. Stevens, *Internetworking with TCP/IP: Principles, protocols, and architecture*, Pearson Prentice Hall, 2006. [Online]. Available: https://books.google.co.in/books?id=jonyuTASbWAC.