A priority queue-based telemonitoring system for automatic diagnosis of heart diseases in integrated fog computing environments

Ali Golkar  
Department of Computer Engineering, Yasooj Branch, Islamic Azad University, Yasooj, Iran

Razieh Malekhosseini  
Department of Computer Engineering, Yasooj Branch, Islamic Azad University, Yasooj, Iran

Keyvan RahimiZadeh  
Department of Computer Engineering, School of Engineering, Yasouj University, Yasouj, Iran

Azita Yazdani  
Department of Health Information Management, Clinical Education Research Center, Health Human Resources Research Center, School of Health Management and Information Sciences, Shiraz University of Medical Sciences, Shiraz, Iran

Amin Beheshti  
Department of Computing, Macquarie University, Sydney, Australia

Abstract
Various studies have shown the benefits of using distributed fog computing for healthcare systems. The new pattern of fog and edge computing reduces latency for data processing compared to cloud computing. Nevertheless, the proposed fog models still have many limitations in improving system performance and patients’ response time. This paper proposes a new performance model by integrating fog computing, priority queues and certainty theory into the Edge computing devices and validating it by analyzing heart disease patients’
conditions in clinical decision support systems (CDSS). In this model, a Certainty Factor (CF) value is assigned to each symptom of heart disease. When one or more symptoms show an abnormal value, the patient’s condition will be evaluated using CF values in the fog layer. In the fog layer, requests are categorized in different priority queues before arriving into the system. The results demonstrate that network usage, latency, and response time of patients' requests are respectively improved by 25.55%, 42.92%, and 34.28% compared to the cloud model. Prioritizing patient requests with respect to CF values in the CDSS provides higher system Quality of Service (QoS) and patients’ response time.

**Keywords**
Fog computing, cloud computing, internet of things, telemonitoring, clinical decision support systems, heart diseases, priority queue

**Introduction**
Recent developments in Information Technology solutions have led to using them broadly in precise health monitoring and treatment systems.¹ Remote patient monitoring (RPM) is the use of digital technologies to monitor and collect patient health data.² RPM systems are used to diagnose medical emergencies, the severity of the patient’s condition, and effective response time to perform medical procedures.³ Internet of things (IoT) has been widely used to join existing medical resources and has provided credible, effective, and intelligent health care services to patients.⁵

Nevertheless, things in the IoT have limited power, storage, and computing capabilities. In this way, cloud computing promises several benefits to deliver cost-effective services. It provides scalable management of virtual storage, processing, and network resources on the Internet and enables ubiquitous data access. Despite the many benefits, responding to latency-dependent and urgent health monitoring and surveillance systems requests is challenging. The main reason for such latency-sensitive applications is the need to perform a huge amount of calculations on large volumes of data and transfer them to a centralized database in data centers.⁶

Introduced by CISCO in 2012, fog computing⁷ has been widely used in the healthcare industry to provide services to applications with urgent response and minimal service delays. Based on predefined policies in this computing model, part of the data collected by the sensors is processed in endpoint IoT devices and the remaining part is sent to the cloud for processing. Fog computing regulates which information should be sent to the cloud server and processed locally. In other words, fog computing as a virtual platform is embedded at the edge of the network, and data processing, storage, and network-based services are moved near to IoT devices instead of data centers.⁸ Low latency services and reduced bandwidth traffic are the main benefits of fog computing. Therefore, fog computing can effectively cope with problems such as interruptions, delays, and even network breakdowns that can have unpleasant consequences for patients’ health in remote health care services. In this way, Clinical decision support systems (DSSs) as interactive information systems can effectively scale to the growth and development of IoT services and data explosion. Nowadays, CDSSs are widely used to predict and diagnose acute clinical conditions based on the data collected by sensors and remote health monitoring devices. These architectures allow real-time analysis of data with minimal latency and response time.⁹ Using fog computing capabilities to implement CDSSs can overcome the lack of awareness, mobility support, and low latency that are existed in the architecture of cloud-based DSS systems.⁵
**Problem statement**

In the current fog models, a mass of data is sent to the cloud server and an important portion of this data contains the vital patients’ health information, which must be processed timely. Moreover, there is no difference between emergency requests and non-emergency requests in receiving services from the cloud server in most such models.

**Research questions**

- Can the use of priority queues in the fog layer lead to improved response time in fog computing compared to cloud computing?
- Can priority queues reduce network usage time in fog computing compared to cloud computing?
- Can priority queues reduce latency in fog computing compared to cloud computing?

**The key contributions**

In this paper, a real-time RPM is presented to continuously monitor and diagnose patients’ health conditions based on the received data from sensors. To increase the efficiency of the proposed system, an innovative model is proposed to classify requests in CDSS based on their priority for receiving the service. According to this model, requests detected as urgent will attain a higher priority for sending to the cloud server. So far, however, there has been little discussion about prioritizing patients’ requests in real-time RPM.

**Related work**

Hassan et al.\textsuperscript{10} presented a home hospitalization system based on IoT, fog computing, and cloud computing. The system allows to manage and monitor the patient’s health vital signs. Azimi et al.\textsuperscript{11} proposed a hierarchical architecture called HiCH is based on fog computing and machine learning for health monitoring. HiCH used the fog and cloud. HiCH efficacy has been evaluated through a comprehensive performance assessment with focusing on arrhythmia detection for patients with cardiovascular disease (CVDs).

In,\textsuperscript{6} a framework called HealthFog is proposed to integrate deep learning in Edge computing devices for automated analysis of heart disease. HealthFog was evaluated for power consumption, network bandwidth, latency, jitter, accuracy, and execution time to deploy and test the performance of the proposed model. He et al.\textsuperscript{12} developed an IoT-based healthcare model called FogCepCare to assess patient health. In this model, partitioning and clustering methods and parallel processing techniques have been used to optimize execution time. FogCepCare performance was compared to the cloud environment in terms of execution time as an optimization parameter. Barik et al.\textsuperscript{13} presented an in-home monitoring system for diagnosing diabetes using k-means clustering methods and deep neural network data on River Ganga Basin geospatial database. Azimi et al.\textsuperscript{14} investigated the possibility of using the convolutional neural network (CNN) classification model as an example of deep learning methods in the fog computing architecture. In this study, a system for monitoring patients’ health using ECG classification is presented. Pham et al.\textsuperscript{15} presented a cloud-based smart home environment, which uses wearable sensors to collect physiological, motor, and acoustic data to periodically monitor patients’ health status. In this study, patients’ daily activities, as well as their behavioral changes, are examined.
Sahoo et al.\textsuperscript{16} proposed a parallel semi-naive bayes (PSNB) based probabilistic method that is used to process large volumes of multi-structured and low-latency patient data in healthcare for future health condition prediction. The optimal data distribution (ODD) algorithm is used to determine the patients’ priorities. Abdelmoneem et al.\textsuperscript{17} introduced a task scheduling and allocation mechanism for cloud-fog based architecture (CFBA) for IoT-based healthcare to monitor the health status of patients. This mechanism performs the task of distributing health care tasks efficiently. Performance evaluation of CBFA was performed based on latency parameters using the iFogSim tool. Gia et al.\textsuperscript{18} developed a health monitoring system that performs continuous remote monitoring of the ECG with automated analysis and notification. ECG sensor nodes collect and transmit respiration rate and body temperature. In addition, the system can analyze the collected data in useful ways to make decisions and, if necessary, send real-time notifications for immediate attention. Gill et al.\textsuperscript{19} proposed an IoT-based cloud computing model for healthcare to effectively manage patients’ heart data. In this model, patients’ status is examined to identify and diagnose heart disease through requests sent by users. The iFogSim toolkit was used to evaluate the proposed model. Mutlag et al.\textsuperscript{9} have developed an electronic health system based on fog calculations for monitoring the health of the elderly. In this system, a mobile application such as the fog node is responsible for periodically collecting patients’ physiological parameters. This system enables to monitor the patient’s health remotely and in real-time.

Muhammed et al.\textsuperscript{20} proposed a healthcare framework called UbeHealth. This study leverages edge computing, deep learning, big data, high-performance computing (HPC), and the IoT to address many challenges that hinder networked healthcare domains. In this study, several algorithms have been used in the form of three layers and two components to predict network traffic to optimize data rates, data storage, and routing decisions. Asif-Ur-Rahman et al.\textsuperscript{21} introduced a heterogeneous cloud-based IoHT communication framework by combining the mist and fog computing in which the routing process for real-time and conventional data is performed separately. The latency criterion has been evaluated to ensure the QoS of the proposed framework. Karakaya et al.\textsuperscript{22} proposed a light and safe fog-based IoT model to report results to the technical team and physicians by analyzing health and tactical information on exercise. This study investigates a fog-based intelligent DSS health care system to automatically diagnose heart patients’ health status based on the nature of patients’ requests.

**Methodology**

Our goal in this study is to introduce an RPM system for heart patients. This system focuses on the early detection of complications that can lead to heart problems in patients. In this study, an end-to-end solution is presented based on a patient health monitoring system. This system aims to help the healthcare team by early detection and continuous monitoring of the patient’s condition and alerting the patient’s suspicious condition. Figure 1 shows the data flow in the proposed model. The model consists of three main parts, including the data layer, fog layer, and cloud layer.

- The data collection layer gathers patients’ data sent from sensors of the body area network (BAN) in real-time and sends it to the fog layer.
- The fog layer stores and evaluates the sensors’ data in a local storage. The data is compared with the normal values ranges (thresholds) predetermined by physicians. If the threshold is violated, the data is sent to the cloud for further evaluation. In the proposed model, requests are prioritized in the fog layer in order to be submitted to the cloud server for CDSS evaluation based on their class (emergency, warning, or normal).
Requests are evaluated to determine the patient’s condition at this layer. The results of the CDSS evaluation are considered as the patient’s condition and if the patient is in a critical or emergency situation, the physician and treatment team, family, and the patient will be informed.

This study uses Raspberry Pi capabilities as a fog node to receive and process sensors data. The Raspberry Pi board is an embedded minicomputer and is very useful in IoT projects. Table 1 represents the specification of the fog node in this study.

**Data preprocessing**

Based on the proposed model in Figure 1, the data sent by the sensors is considered as request demands. Each request demand includes various vital signs of the patient that are sent to the fog node. For each symptom different ranges including normal, high, very high, low, and very low are defined by medical professionals. Table 2 shows typical ranges defined by expert physicians according to the importance and level of risks for the patients. This quantification is based on certainty factors (CF) and the effective ratio of each symptom (Table 4) in heart disease.23

Certainty factors is a technique for uncertainty in expert systems and expresses the level of belief in an event, a fact, or a hypothesis using expert assessment.24 The CF model has become a standard way to manage uncertainty in rule-based systems. CF theory is an alternative to Bayesian reasoning when reliable statistical information is not available or the independence of evidence cannot be assumed. This theory introduces a certainty factors calculus based on human expert heuristics. According to this approach, as shown in Table 2, uncertain statements are mapped to a CF number between 1 and −1, which indicates the degree of belief in a statement. A certainty factor
A factor of 0 would represent a complete lack of belief in something while a factor of 1 would represent an absolute belief in a rule or value.

For example, in the sentence “it will probably rain today”, CF (E) = CF (it will probably rain today) = 0.6. This approach replaces the more formal probability P (E) with CF (E).

Since the effective symptoms in the diagnosis of heart disease have been used, the range of [0–1] has been used for mapping to the CF number. According to Table 2 and based on the symptoms of heart disease and the opinion of medical experts, Table 3 is prepared to evaluate the symptoms of heart disease. When one or more symptoms of a request are outside the normal range, the request will be weighted using Table 3 for prioritization. CF values will be awarded to all ranges of each symptom by physicians using Table 3. This CF value indicates the effect of the measured value of each symptom on heart disease. For example, CF value for the very high and very low ranges of the blood pressure symptom is 0.89 and 0.55, respectively. High blood pressure is one of the most important risk factors for CVD and is the leading cause of mortality.\textsuperscript{25}

When a request \(i\) is sent to the fog node, all \(k\) abnormal symptoms are aggregated according to equation (1).

\[
    w_i = \sum_{k=0}^{n} \text{CF(Abnormal Symptom}_k\text{)}
\]  

Equation (2) normalizes the weight of request \(i\), \(\pi_i\), into range [0,1].

\[
    \text{weight(request}_i\text{)} = \frac{(w_i - \min(\text{Abnormal Symptom}_k)) \times (1 - \varepsilon)}{(w_i - \min(\text{Abnormal Symptom}_k) + \varepsilon)}
\]
According to the proposed model in the fog layer, a request containing one or more abnormal symptoms is detected and is sent to the cloud layer. However, when the number of these requests increases the queue length increases. The queuing service in the proposed model prioritizes the process of requests based on the urgency of the demand. Each request is weighted based on the number of abnormal symptoms and CF values. Various methods are used to manage requests queue. One of the most popular is the M/G/1 queueing model with priority. Inter arrival times of submitted requests to the system are exponentially distributed and the service time distribution can be any arbitrary distribution. In this queueing model, the requests are classified into different static priorities before entering the system. A request is assigned to a waiting queue upon arrival based on its priority class. Priority classes are numbered from one to h. h is the highest priority class and one is assigned to the lowest priority class. In the proposed model, the request weight calculated by equation (2) is assumed as the priority of that request. Higher priority is assigned to patients’ requests with urgent demand. The response time of a request with priority \( p \) \((p = 1, 2, \ldots, h)\), \( T_p \), is the sum of average request service time of class \( p \) requests, \( E[S] \), and the average time that an arriving request of class \( p \) has to wait, \( W_p \), for the server to complete processing the current request in service, equation (3).

\[
T_p = E[S] + W_p
\]  

The request waiting time is a calculated as follows.

\[
W_p = \frac{1}{2} \sum_{i=1}^{h} \lambda_i \times E[S_i^2] \quad \frac{1}{(1 - U_p) \times (1 - U_{p+1})}
\]  

where, \( p, \lambda_p, E[S_p^2] \), and \( U_p \) denote priority class, arrival rate of requests of priority \( p \), the second moment of class \( p \) requests, and is the server utilization due to processing class \( p \) requests, respectively. \( U_p \) is calculated by equation (5).

\[
U_p = \sum_{j=p}^{h} \lambda_j \times E[S_j]
\]  

The product \( \lambda_j \times E[S_j] \) is the fog server utilization \( U_j \) due to processing priority \( j \) requests.

In this study, patients’ requests are classified to three classes named normal, warning, and emergency classes after registering in the electronic health record (EHR). The emergency requests have the highest priority and the normal requests classes have the lowest priority for processing in the fog layer (Figure 2). The requests with the warning class have the middle priority.
Results

Dataset

The coronary artery heart disease patient database in the machine learning repository of the University of California, Irvine \(^{27}\) is used in experiments. In this dataset, 14 different characteristics are measured. It includes the characteristics of 320 samples of people with coronary artery disease.

Evaluation

A simulator (mobile application) was developed to evaluate the proposed model. This mobile application was developed using Android Studio to send the patient’s vital signs from the data-acquisition layer to the fog layer. It sends dataset records as requests based on the Poisson distribution function with arrival rate \(\lambda\). Data attributes are stored separately in a CSV file and are uploaded to the broker node using HTTP POST. Requests are sent by this application in real-time to the fog layer to monitor the patient’s health condition.

Main three component of Figure 3 are:

A: An android application that sends dataset records to the Raspberry Pi. This application simulates the role of an IoT device and sends requests to the cloud server for evaluation. Each record contains a number of features.

B: As a fog layer, Raspberry Pi is responsible for classifying the requests sent by the android application. According to Table 4, each request contains 14 features that are considered as symptoms measured by sensors.

C: The cloud layer contains a clinical DSS that is responsible for interpreting the requests sent from the fog layer. Microsoft Azure cloud platform is used in this layer.
The CDSS used in the cloud layer is based on the rule developed in the study. Some samples of the extracted rules are shown as follows:

- **R1**: if (thal is 3) and (ca is 3) and (sex is 0) and (slope is 1) then the probability of infection is 56%.

- **R2**: if (thal is 6) and (CP is 4) and (restecg is 1) and (sex is 0) and (chol >200) then the probability of infection is 64%.

- **R3**: if (thal is 3) and (ca is 3) and (sex is 1) and (slope is 2 or 3) and (thalach ≤ 160) then the probability of infection is 5%.

Figure 4 shows the measured network usage time for different requests arrival rates. As can be seen, the average network usage time in the proposed model is less than in the fog and cloud-based model. In our model, requests detected as normal by the fog node are stored locally, so the warning and emergency requests classes need to be checked in real-time and transmitted through the network first. Normal class requests are sent to the cloud layer to record data in the EHR when the queue of warning and emergency classes is empty. The proposed fog-based model reduced network usage time by an average of 25.55% compared to the cloud compared and 5.8% to the fog computing model.

We present the aggregated results for all the classes in terms of Latency and Response Time in Figures 5 and 6. In this way, we used the following calculations for these metrics

\[
L_{\text{aggr}} = \frac{\sum_{i=1}^{j} (w_{\text{class}_i} \times L_{\text{class}_i})}{\sum_{i=1}^{j} w_{\text{class}_i}}
\]  

(6)
where, $L^{aggr}$ and $R^{aggr}$ denote the aggregated (weighted) latency and response time of a request in an infrastructure (Cloud, Fog, or Proposed). $w_{class_i}$ represents the weight of the request class $i$ (as shown in Table 5), $L_{class_i}$ represents the latency of class $i$, and $R_{class_i}$ represents the response time of class $i$. There are three classes ($j = 3$) including Normal, Warning, and Emergency.

Figure 5 depicts a comparison of the average latency in the proposed model and fog computing and cloud computing with different patient requests rates. Compared to cloud and fog computing, the proposed model reduced the latency by an average of 42.92% and 14.5% respectively.

Figure 6 represents a comparison of the average response time in the proposed model with the fog and cloud computing based on equation (3). Compared to cloud and fog computing, the proposed model reduced the response time by an average of 34.28% and 15.6% respectively.
Table 5 shows the classification of dataset records in the fog layer. In this layer, the request classification process was performed by equation (2).

These results demonstrate that the proposed model provides better performance in comparison to the others in lowering the average latency and response time of a request.

![Figure 4. Average network usage time versus patients’ request rates.](image1)

![Figure 5. Average request latency versus patients’ request rates.](image2)

Table 5 shows the classification of dataset records in the fog layer. In this layer, the request classification process was performed by equation (2).

These results demonstrate that the proposed model provides better performance in comparison to the others in lowering the average latency and response time of a request.
Figure 7 shows a comparison of the latency in the proposed model with the cloud and fog model for different patient request classes. According to this figure, the proposed model reduced the requests latency of warning classes 45.53% and 54.5% respectively compared to the fog and cloud model. Also requests latency of emergency classes 19.7% and 32.02% respectively reduced compared to the fog and cloud model.

Figure 8 represents a comparison of network usage time in the proposed model with the cloud and fog model for three requests classes. As can be seen, warning class requests consumed more network bandwidth than emergency and normal classes because the main portion of requests is warning. The proposed model reduced the network usage time for requests in warning class 26.39% and 14.46% respectively reduced compared to the cloud and fog model. Also emergency class network usage time 15.66% and 8.74% reduced compared to the cloud and fog model.

Figure 9 shows the comparison of response time in the proposed model. According to this figure, emergency class requests have a better response time than the other two classes. The proposed model improved the response time of warning in fog model 18.15% and cloud model 26.94, also 61.41% and 77.34% respectively reduced for emergency class in fog and cloud model.

The results in Figures 7 and 9 represent the Latency and Response Time for each class. With respect to Normal class results in these Figs., it is necessary to mention that the DSS in our model detects a request class firstly and processes the request based on its class type. All the requests experience a processing time in the DSS. Therefore, the DSS imposes a negligible time to the

**Table 5.** Requests classification result.

| Class    | Number of requests | Percentage, % |
|----------|--------------------|---------------|
| Normal   | 100                | 31            |
| Warning  | 132                | 41            |
| Emergency| 88                 | 28            |

Figure 7 shows a comparison of the latency in the proposed model with the cloud and fog model for different patient request classes. According to this figure, the proposed model reduced the requests latency of warning classes 45.53% and 54.5% respectively compared to the fog and cloud model. Also requests latency of emergency classes 19.7% and 32.02% respectively reduced compared to the fog and cloud model.

Figure 8 represents a comparison of network usage time in the proposed model with the cloud and fog model for three requests classes. As can be seen, warning class requests consumed more network bandwidth than emergency and normal classes because the main portion of requests is warning. The proposed model reduced the network usage time for requests in warning class 26.39% and 14.46% respectively reduced compared to the cloud and fog model. Also emergency class network usage time 15.66% and 8.74% reduced compared to the cloud and fog model.

Figure 9 shows the comparison of response time in the proposed model. According to this figure, emergency class requests have a better response time than the other two classes. The proposed model improved the response time of warning in fog model 18.15% and cloud model 26.94, also 61.41% and 77.34% respectively reduced for emergency class in fog and cloud model.

The results in Figures 7 and 9 represent the Latency and Response Time for each class. With respect to Normal class results in these Figs., it is necessary to mention that the DSS in our model detects a request class firstly and processes the request based on its class type. All the requests experience a processing time in the DSS. Therefore, the DSS imposes a negligible time to the
Latency and Response time of all requests. This is the main reason that the Normal requests face more Latency and Response time in our model in comparison to the Fog and Cloud.

Figures 7 and 9, show that when the proposed model detects a request as normal, the request would not be processed until all the requests in the higher priority classes processed. It is clear from

Figure 7. Comparison of latency time in each classes.

Figure 8. Comparison of network usage time in each classes.

Latency and Response time of all requests. This is the main reason that the Normal requests face more Latency and Response time in our model in comparison to the Fog and Cloud.

Figures 7 and 9, show that when the proposed model detects a request as normal, the request would not be processed until all the requests in the higher priority classes processed. It is clear from
equations (3)–(5) that the waiting time of a lower class requests (normal) depends on the number of higher priority queues (warning and emergency) and the number of requests in these queues. When the DSS in our model receives a request, it needs a time to detect the request class based on its characteristics. Therefore, the DSS imposes a negligible time to the Latency and Response time of all requests. This is a reason that normal requests face more Latency and Response time in our model in comparison to the Fog and Cloud. Therefore, the normal requests relatively experience high latency and response time in comparison to the fog and cloud environments. When a request is a normal request, there is no sensitivity to send it in a dedicated time in our model. This would not affect the functionality of the proposed model.

Discussion

In this study, by using priority queues in the fog layer and allocating appropriate bandwidth to the requests of patients with acute conditions, a significant improvement in response time (34.28%) was achieved. In recent research, using techniques such as big data, decision support systems, machine learning, etc., in the fog layer, attempts have been made to manage bandwidth. Fog-based remote patient monitoring systems have mainly focused on latency improvement.

Integrating the IoT, fog computing, and cloud computing has provided many benefits, such as remote patients’ treatment plans and urgent medical care for healthcare systems. Recently, the efficiency of cloud computing and its use in the medical industry has been extensively studied, leading to the provision of efficient and accepted cloud-based solutions. However, due to geographically centralized architecture and multi-stage distance from the IoT data sources, the cloud faces limitations in real-time communications. The lack of real-time communication can increase the number of patients’ death in IoT-based healthcare systems.

Table 6 examines recent works in the fog-IoT-based healthcare industry. Previous works has focused mainly on the analysis of cardiac patient data and predictive accuracy, and less attention has been paid to the QoS review of RPMs. The presented model in this study can effectively identify the
Table 6. The percentage of improvement of the proposed model compared to the other models.

| Work        | Fog computing | IoT | Prediction system | Priority queue | Power consumption | Execution time | Latency | Network usage time | Response time |
|-------------|----------------|-----|-------------------|----------------|-------------------|----------------|---------|-------------------|---------------|
| [10]        | ✔              | ✔   |                   |                | —                 | —              | —       | —                 |               |
| [11]        | ✔              | ✔   |                   |                | —                 | —              | —       | —                 | 16.45%        |
| [6]         | ✔              | ✔   | 89%               | —              | —                 | —              | —       | 16%               | 26%           |
| [12]        | ✔              | ✔   |                   |                | —                 | —              | —       | —                 |               |
| [13]        | ✔              | ✔   | 81.89%            | —              | —                 | —              | —       | —                 |               |
| [14]        | ✔              | ✔   | 96%               | —              | —                 | 72.8%          | —       | —                 | 19.04%        |
| [15]        | ✔              | ✔   | 90%               | —              | —                 | —              | —       | —                 |               |
| [16]        | ✔              | ✔   | 87.8%             | —              | —                 | —              | —       | —                 |               |
| [17]        | ✔              | ✔   |                   | —              | —                 | —              | —       | —                 |               |
| [18]        | ✔              | ✔   | —                 | 20–50%         | —                 | 10–50%         | 40–80%  | —                 |               |
| [19]        | ✔              | ✔   | —                 | 23.56%         | —                 | 19.5–22%       | 22–26.7%| —                 | 29.4%         |
| [22]        | ✔              | ✔   | ✔                 | —              | —                 | 1%             | 35.4%   | 11.16%            | —             |
| Proposed model | ✔              | ✔   | 89.76%            | ✔              | —                 | —              | 42.92%  | 25.55%            | 34.28%        |
requests based on the severity of the patient’s request. It determines the priority of requests by identifying the symptoms affecting the condition of heart patients in fog layer. In the proposed model, emergency requests are prioritized for receiving services by combining the weighting technique and priority queues.

Recent studies have presented models to reduce latency and make applications more efficient using fog and cloud computing. In this study, the effectiveness of using priority queues to identify emergency patients with minimal delay was investigated. According to Table 6, although the response time criterion is an important factor in the provision of remote patient monitoring services, most of the fog-based RPM models have qualitatively an improvement in response time reported. The proposed model, unlike previous studies, is simultaneously focused on increasing the quality of service and the quality of experience (QoE).

In the evaluation of the present model, the input data is based on a dataset, which as a limitation, vital signals can be directly received from sensors in the real environment.

Conclusion

In this study, a fog-based intelligent DSS health care system was presented for automatic diagnosis of heart patients’ health status. This health care system introduces a queuing modeling in the fog service to effectively evaluate heart patients’ data received from various IoT devices. The system incorporates the queuing modeling into edge devices for real-time heart disease analysis and response. The proposed fog computing model prioritizes the incoming patients’ requests with different medical urgencies to be processed by the DSS. The proposed system was evaluated with the coronary artery heart disease dataset and the results demonstrate that it can significantly improve QoS metrics. Moreover, this system scales well when the request rate increases. In the future, we plan to investigate a method to switch requests class between queues. In other words, the CF-based mechanism is tuned based on the waiting time of requests and move them between classes. In addition, the efficiency of the proposed system can be evaluated for predicting and diagnosing other IoT-based diseases.

Author’s NoteThis article is extracted from the doctoral thesis.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

Ethical approval

All the data in this study have been downloaded from the Heart Disease Data Set available online in the UCI Machine Learning Repository. This database provides public and open access for researchers. Since the present study was conducted using de-identified data, this study was also exempted from ethics review by the Islamic Azad University, Yasooj branch (Number: 162442797, Date: 2021-July-12).
References

1. Kalid N, Zaidan AA, Zaidan BB, et al. Based real time remote health monitoring systems: a review on patients prioritization and related “big data” using body sensors information and communication technology. J Med Syst 2018; 42(2): 30.
2. Vegesna A, Tran M, Angelaccio M, et al. Remote patient monitoring via non-invasive digital technologies: a systematic review. Telemed E-Health 2017; 23(1): 3–17.
3. Taiwo O and Ezugwu AE. Smart healthcare support for remote patient monitoring during covid-19 quarantine. Inform Med Unlocked 2020; 20: 100428.
4. Sapci AH and Sapci HA. Innovative assisted living tools, remote monitoring technologies, artificial intelligence-driven solutions, and robotic systems for aging societies: systematic review. JMRI Aging 2019; 2(2): e15429.
5. Albahri OS, Albahri AS, Mohammed KI, et al. Systematic review of real-time remote health monitoring system in triage and priority-based sensor technology: taxonomy, open challenges, motivation and recommendations. J Med Syst 2018; 42(5): 1–27.
6. Tuli S, Basumatory N, Gill SS, et al. HealthFog: an ensemble deep learning based smart healthcare system for automatic diagnosis of heart diseases in integrated IoT and fog computing environments. Future Gen Comput Syst 2020; 104: 187–200.
7. Cisco. Fog Computing and the Internet of Things: Extend the Cloud to Where the Things Are, [White paper]. (accessed 21 March 2022).
8. Hu P, Dhelim S, Ning H et al. Survey on fog computing: architecture, key technologies, applications and open issues. J Network Comput Appl 2017; 98: 27–42.
9. Mutlag AA, Abd Ghani MK, Arunkumar N, et al. Enabling technologies for fog computing in healthcare IoT systems. Future Gener Comput Syst 2019; 90: 62–78.
10. Hassen HB, Ayari N and Hamdi B. A home hospitalization system based on the internet of things, fog computing and cloud computing. Inform Med Unlocked 2020; 20: 100368.
11. Azimi I, Anzanpour A, Rahmani AM, et al. HiCH: hierarchical fog-assisted computing architecture for healthcare IoT. ACM Trans Embed Comput Syst (Tecs) 2017; 16(5s): 1–20.
12. He S, Cheng B, Wang H, et al. Proactive personalized services through fog-cloud computing in large-scale IoT-based healthcare application. China Commun 2017; 14(11): 1–16.
13. Barik RK, Priyadarshini R, Dubey H, et al. FogLearn: leveraging fog-based machine learning for smart system big data analytics. Int J Fog Comput (IJFC) 2018; 1(1): 15–34.
14. Azimi I, Takalo-Mattila I, Anzanpour A, et al. Empowering healthcare IoT systems with hierarchical edge-based deep learning. In: Proceedings of the 2018 IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies, Washington, DC, 26–28 September 2018.
15. Pham M, Mengistu Y, Do H, et al. Delivering home healthcare through a cloud-based smart home environment (CoSHE). Future Gener Comput Syst 2018; 81: 129–140.
16. Sahoo PK, Mohapatra SK and Wu S-L. SLA based healthcare big data analysis and computing in cloud network. J Parallel Distrib Comput 2018; 119: 121–135.
17. Abdelmoneem RM, Benslimane A, Shaaban E, et al. A cloud-fog based architecture for iot applications dedicated to healthcare. In: ICC 2019-2019 IEEE International Conference on Communications (ICC), Shanghai, China, 20–24 May 2019. New York: IEEE, 2019.
18. Gia TN, Jiang M, Sarker VK, et al. Low-cost fog-assisted health-care IoT system with energy-efficient sensor nodes. In: 13th International Wireless Communications and Mobile Computing Conference (IWCMC), Valencia, Spain, 26–30 June 2017. New York: IEEE, 2017.
19. Gill SS, et al. Fog-based smart healthcare as a big data and cloud service for heart patients using IoT. In: *International conference on intelligent data communication technologies and internet of things*. Cham: Springer, 2018.

20. Muhammed T, Mehmood R, Albeshri A, et al. UbeHealth: a personalized ubiquitous cloud and edge-enabled networked healthcare system for smart cities. *IEEE Access* 2018; 6: 32258–32285.

21. Asif-Ur-Rahman M, Afsana F, Mahmud M, et al. Toward a heterogeneous mist, fog, and cloud-based framework for the internet of healthcare things. *IEEE Int Things J* 2018; 6(3): 4049–4062.

22. Karakaya A and Akleyelek S. A novel IoT-based health and tactical analysis model with fog computing. *PeerJ Comput Sci* 2021; 7: e342.

23. Widians JA, Puspitasari N and Febriansyah A. Disease diagnosis system using certainty factor. In: International Conference on Electrical, Electronics and Information Engineering (ICEEIE), Denpasar, Indonesia, 03–04 October 2019. New York: IEEE, 2019.

24. Shortliffe EH and Buchanan BG. A model of inexact reasoning in medicine. *Math Biosci* 1975; 23(3–4): 351–379.

25. He J and Whelton PK. Elevated systolic blood pressure and risk of cardiovascular and renal disease: overview of evidence from observational epidemiologic studies and randomized controlled trials. *Am Heart J* 1999; 138(3): S211–S219.

26. Kleinrock L. *Computer applications*. Wiely-Intersciense, Queueing Systems, 1976.

27. Blake C, Keogh E and Merz CJ. *UCI repository of machine learning databases*. Irvine, CA: Department of Information and Computer Science, University of California, 1998, [http://www.ics.uci.edu/~mlearn/MLRepository.html](http://www.ics.uci.edu/~mlearn/MLRepository.html)

28. Zipes DP, Libby P, Bonow RO, et al. *Braunwald’s heart disease e-book: a textbook of cardiovascular medicine*. London, UK: Elsevier Health Sciences, 2018.

29. Wang L. Intelligent decision support system of emergency language based on fog computing. *Math Probl Eng* 2021; 2021: 6611501.