electronics

MDPI

Article

Intelligent Fog-Enabled Smart Healthcare System for Wearable Physiological Parameter Detection

Muhammad Ijaz [1,2], Gang Li [1,2], Huiquan Wang [3], Ahmed M. El-Sherbeeny [4], Yussif Moro Awelisah [1,2], Ling Lin [1,2,*], Anis Koubaa [5,6] and Alam Noor [5,*]

[1] State Key Laboratory of Precision Measurement Technology and Instrument, Tianjin University, Tianjin 300072, China; ijazm09@gmail.com (M.I.); ligang59@tju.edu.cn (G.L.); yussif@tju.edu.cn (Y.M.A.)
[2] China and Tianjin Key Laboratory of Biomedical Detecting Techniques and Instruments, Tianjin University, Tianjin 300072, China
[3] School of life sciences, Tiangong University, Tianjin 300387, China; huiquan@tiangong.edu.cn
[4] Industrial Engineering Department, College of Engineering, King Saud University, PO Box 800, Riyadh 11421, Saudi Arabia; aelsherbeeny@ksu.edu.sa
[5] Robotics and Internet-of-Things Lab, Prince Sultan University, Riyadh 11586, Saudi Arabia; akoubaa@psu.edu.sa
[6] CISTER, INESC-TEC, ISEP, Polytechnic Institute of Porto, 4249-015 Porto, Portugal
* Correspondence: linling@tju.edu.cn (L.L.); eng.alamnoor@gmail.com (A.N.)

Received: 26 October 2020; Accepted: 24 November 2020; Published: 28 November 2020

Abstract: Wearable technology plays a key role in smart healthcare applications. Detection and analysis of the physiological data from wearable devices is an essential process in smart healthcare. Physiological data analysis is performed in fog computing to abridge the excess latency introduced by cloud computing. However, the latency for the emergency health status and overloading in fog environment becomes key challenges for smart healthcare. This paper resolves these problems by presenting a novel tri-fog health architecture for physiological parameter detection. The overall system is built upon three layers as wearable layer, intelligent fog layer, and cloud layer. In the first layer, data from the wearable of patients are subjected to fault detection at personal data assistant (PDA). To eliminate fault data, we present the rapid kernel principal component analysis (RK-PCA) algorithm. Then, the faultless data is validated, whether it is duplicate or not, by the data on-looker node in the second layer. To remove data redundancy, we propose a new fuzzy assisted objective optimization by ratio analysis (FaMOORA) algorithm. To timely predict the user's health status, we enable the two-level health hidden Markov model (2L-2HMM) that finds the user's health status from temporal variations in data collected from wearable devices. Finally, the user's health status is detected in the fog layer with the assist of a hybrid machine learning algorithm, namely SpikQ-Net, based on the three major categories of attributes such as behavioral, biomedical, and environment. Upon the user's health status, the immediate action is taken by both cloud and fog layers. To ensure lower response time and timely service, we also present an optimal health off procedure with the aid of the multi-objective spotted hyena optimization (MoSHO) algorithm. The health off method allows offloading between overloaded and underloaded fog nodes. The proposed tri-fog health model is validated by a thorough simulation performed in the iFogSim tool. It shows better achievements in latency (reduced up to 3 ms), execution time (reduced up to 1.7 ms), detection accuracy (improved up to 97%), and system stability (improved up to 96%).

Keywords: Tri-Fog Health System; fault data elimination; health status prediction; health status detection; health off

1. Introduction

In recent times, smart healthcare becomes an emerging application of the internet of things (IoT). A smart healthcare system consists of wearable sensors used to monitor the specific health status of the users or patients [1,2]. Most importantly, wearable technology has become a vital part of not only remote patient monitoring but also for user health monitoring regularly. The introduction of wearable devices minimizes the frequent involvement of doctors in health monitoring. It also assists in the early detection of diseases, drug research, smart hospital development, and safety provisioning [3,4]. Two major technologies have to be investigated to develop a smart healthcare system. Firstly, through biomedical sensors like temperature, motion, blood pressure, and how the wearable devices are appended on the user's body to acquire, their health status is studied [5]. Secondly, the useful fog computing technology that enables real-time and delay minimized health services [6,7] should be examined. As a result of the fog enabled healthcare system, the data acquired from the wearable devices are analyzed by the fog layer, which further minimizes the latency for healthcare services.

In the wearable based physiological monitoring, fault data detection from biosensors is one of the main issues [8]. Analyzing the fault data sensed by the wearable leads to inaccurate decision making. As the health data must be more reliable, the adaptive neuro-fuzzy inference system (ANFIS) is presented to diagnose health data [9]. A distributed similarity test has been introduced to detect the sensor data fault [10]. Based on the sensor reading variations, the sensor fault is detected. Thus, in a smart healthcare system, fault data diagnosis and elimination are the first and foremost process. In recent times, deep learning is addressed as one of the effective solutions for classification and prediction problems [11,12]. Deep learning algorithms are capable of handling and analyzing the massive amount of data without minimum prediction errors [13,14]. When it comes to fog-enabled smart healthcare, the faultless data need to be analyzed by the fog nodes. In most fog-based smart healthcare systems, a deep learning approach is incorporated to detect the abnormalities in the user's health status [15,16]. The deep learning approaches analyze the user's physiological data aggregated from the wearable devices. The multi-classifier system works based on data fused from different kinds of wearable sensors [17]. Both data fusion and multi-classifier methods offer better performance. A transfer learning model is proposed to detect Parkinson's disease from the wearable data [18].

In conventional healthcare systems, the data are analyzed in the cloud layer, which means the latency is high. Thus, fog computing becomes an essential part of the smart healthcare system [19]. Although the smart healthcare system has potentiality worked in many applications; still, the fault data detection has not yet been addressed. Firstly, while all works focus on the data analysis in the fog layer, the major issue of fog computing (i.e., fog overloading) is not yet concentrated for the healthcare environment. With the increase in users or patients in the healthcare system, there are relatively increasing concerns about the overloading issue [20]. Secondly, most research works utilized only biosensors or body sensors or wearable data to monitor the user's health status. However, environmental factors are also equally important to detect the user's health status accurately [21]. Environmental data like environment temperature, noise level, pollution level have a direct impact on the user's health data generated by wearable devices [22]. Both of the issues, including fog offloading and environmental data analysis, are not thoroughly analyzed in the prior research works from the perspective of healthcare systems. Apart from these issues, another challenge is that the duplicate data processing in fog or cloud layer leads to unnecessary resource consumption [23]. For example, the sensors report the data for each fraction of seconds. But the temperature data may be the same for a while. Thus, redundant data elimination will speed up the analysis process and also preserve the resources of fog and cloud computing.

1.1. Motivation and Inspiration

This research work is motivated by the existing problems in the smart healthcare system, such as high latency for emergency health status, fault and duplicate data analysis, and fog overloading. The brief summarization on research issues in our smart healthcare is provided here:

- Latency—High latency is introduced due to data processing in the cloud layer. In general, healthcare services need on-time data transmission. Latency in healthcare applications directly affects the patient's or user's health.
- Fault Data—Most wearable or biosensors are deployed in the open or mobile (on the human) environment. It makes the sensors to generate erroneous data. Analyzing the inaccurate data ultimately leads to incorrect prediction.
- Low Accuracy—Health status prediction accuracy is down due to improper algorithms and attributes. All research works considered only wearable sensor data, which are unable to provide accurate results. However, environmental data are also crucial for accurate prediction. As the wearable data consists of duplicate data, it also affects the data processing accuracy.
- Fog Overloading—One of the best solutions for minimizing latency is to use fog computing. However, when more users are connected in the same region, the fog will become overloaded. This overloading problem is the major research issue in smart healthcare.

These research issues motivate us to design a novel smart healthcare system. This paper concentrates on all these aspects to design an efficient smart healthcare system. For those, we formulate the following research objectives:

- Design a family healthcare system by using wearable for monitoring physiological parameters.
- Accurately predict the user state through physiological parameters.
- Minimize latency and detect the user state promptly.

To achieve these objectives, we designed a novel tripartite fog-enabled healthcare (Tri-FogHealth) architecture to monitor the health status of users through wearable physiological parameters. Apart from this, we made the following contributions in Tri-FogHealth architecture:

- Erroneous data from the wearable is diagnosed by using a rapid kernel principal component analysis (RK-PCA) algorithm. The RK-PCA algorithm is incorporated in a personal data assistant (PDA) (i.e.,) the faulty data is detected and eliminated in the wearable layer.
- From the faultless data, the data deduplication process is carried by the data on-looker node. To detect and eliminate duplicate data, we present a fuzzy assisted objective optimization by ratio analysis (FaMOORA) algorithm. If no duplicate data is found, then the fog node predicts the user's health status through temporal features using the two-level health hidden Markov model (2L-2HMM). The 2L-2HMM appends the severity level of the user's health status and provides apt action for severe cases.
- To prevent the fog layer from overloading the issue, we enable the health off procedure. The health of works upon two states. In the first stage, the offloading decision is made based on two rules. Then, an optimal node is selected for offloading by using the multi-objective spotted hyena optimization (MoSHO) algorithm.
- Then, the user's current health status is detected in the cloud layer based on behavioral, biomedical, and environmental attributes. All these attributes are fed into the spiking quantum neural network (SpikQ-Net) that accurately classifies the user's health status into normal action required, no action required, and immediate action required classes.

1.2. Paper Layout

The rest of this paper is organized as follows: Section 2 surveys existing research works held on smart healthcare systems. In Section 3, we highlight the problem statement and the background of the problems considered in this work. Section 4 clearly explains the overall proposed work with all algorithms. In Section 5, we describe the experimentation detail of the proposed work with experimental setup and comparative analysis. Section 6 concludes the overall contributions of this work.

2. Related Work

In this section, we present existing important research works proposed in the smart healthcare system.

2.1. Research Works on Cloud-Based Smart Healthcare System

A smart healthcare system is designed to monitor elderly peoples and optimal access control [24]. For providing access control, the Pearson correlation coefficient (PCC) algorithm is presented with the Manhattan distance measure. Firstly, algorithms work in parallel to identify older people. Then, the PCA algorithm is applied to reduce the dimensionality reduction for the collected elderly data. Lately, the normalization cross-correlation (NCC) algorithm is utilized for final results. However, the PCA and NCC algorithms are insufficient to process the large amount of data collected from elderly peoples. In a smart healthcare system designed with multiple heterogeneous wearable sensors [25], the proposed approach uses an analytical hierarchy process (AHP) with the VIseKriterijumska Optimizacija I Kompromisno Resenje (VIKOR) method to make a proper decision on hospital selection based on two steps. Firstly, the data from the wearable device was collected and analyzed. Then, the hospital is selected by considering the user's health status and multiple criteria. In this work, the processing algorithm is only applied to determine the optimal hospital; however, a more intelligent mechanism will make decisions based on the health data.

For reliable medical monitoring, fault data detection was performed by the threshold tuning method [26]. The medical data is collected from wearable devices. As it could be corrupted from the environmental factors and sensor faults, the fault detection using Bayesian neural network was performed initially. Although fault data detection improves data reliability, health status monitoring performance is still inefficient. The Bayesian neural network increases the complexity of the personal devices of the patient.

As a result, a statistical approach was presented for patient monitoring [27]. The vital signs are acquired from the wearable, and the data is processed to predict the patient's health status. First, gaussian process (GP) models are built for interpretive signs with varying complexity. For this purpose, the covariance kernels and the fixed parameters are utilized. Then, the Bayesian model is proposed to find the health status of the user based on two main hypotheses, such as (i) use of patient-specific model and (ii) use of optimal hyperparameter values. Here, the selection of hyperparameters increases the complexity of the system.

A new cloud digital twins (CloudDTH) framework was introduced for smart healthcare monitoring [28]. The concept of digital twins is integrated multi-physics, multi-scale and probabilistic simulation of a system. In CloudDTH, a virtual object is created for all users in the cloud environment. The virtual object data was used for modelling, simulating, and evaluating the user data. From the analysis of the virtual object, the healthcare status of the user was predicted. In general, digital twins are hard to deploy and manage. In addition, to create a digital twin for each user is not scalable.

To further enhance the prediction accuracy, a complement naïve Bayesian (CNB) classifier was deployed in the cloud environment to classify the user's health data [29]. Firstly, the medical data was acquired from the wearable and biosensors. Before classification, the preprocessing mechanism was applied to handle the noisy data. The CNB classifier uses double learning processes to improve classification accuracy. First, the naïve Bayes classifier was trained with the dataset. Then, the classifier was tested by using the same dataset. It is known as double learning. Here the misclassified data was used for original test data classification.

2.2. Research Works on Fog-Enabled Healthcare System

The concept of edge computing is introduced to mitigate latency in healthcare applications [30]. The proposed system computed the criticality measure index (CMI) from the gathered wearable data. Then the alert message was triggered in emergencies. Here, the CMI value is calculated for each data without considering temporal and environmental data. It tends to inaccurate health status prediction.

The fog computing-based gateway design explores a geo-distributed intermediary layer of intelligence [31]. This layer was introduced between the sensors and the cloud. In the fog enabled healthcare systems, the fog node takes the responsibility of the sensors such as severity detection. The highlight is that the use of the fog layer in the e-healthcare system improves energy efficiency, scalability, and reliability problems and also supports user mobility. Therefore, fog computing plays a pivotal role in e-healthcare applications.

However, the data analysis accuracy is low since there no intelligence algorithm is applied to the collected health data. Therefore, a smart treatment for personalized healthcare (STPH) model was presented with edge computing [32]. The STPH model provides the optimal treatment solution to minimize the cost of the intelligent agent. This approach uses an intelligent agent (IA), which assists in constructing a personalized treatment plan. Furthermore, the edge servers were introduced to help the cloud server in the data processing. Here the treatment plan is optimally selected for emergencies. However, optimal treatment plan selection without detecting the user's current health status results in low accuracy.

An edge of thins (EoT) driven ambient assisted living framework was proposed for e-healthcare applications [33]. This framework uses a hybrid classifier to predict the health status of the user. The hybrid classifier is made up of the hybrid ambient assisted living with naïve bayes classifier and the firefly algorithm (HAAL-NBFA). The firefly algorithm is employed to select the optimal features for classification. Upon optimal features chosen by the firefly algorithm, the naïve Bayes classifier detects the state of the user. Before the imbalanced dataset is balanced by the synthetic minority over-sampling technique (SMOTE), the naïve Bayes classifier has less accuracy due to the class dependencies. This work fails when the incoming data has faulty reading.

A HealthFog framework was developed to monitor the heart diseases in the IoT environment to minimize the latency and response time for personalized healthcare [34]. The HealthFog integrates the ensemble deep learning algorithm with edge computing. Further, the fog-enabled framework is defined as FogBus, which efficiently detects the health status of heart patients. At the same time, real-time medical data is collected from the sensors of the wearable. The FogBus is designed using the broker node, the worker node, and the cloud data center. The same data is analyzed in different worker nodes, and the results are ensemble for final classification. Here the deep learning model is trained in each worker node of each fog node; this increases the time consumption.

As fog overloading increases the latency for the emergency data, an ubiquitous cloud edge enabled healthcare (UbeHealth) system was aimed to improve the quality of service (QoS) in the healthcare system [35]. The healthcare data was processed in both the cloudlet and network layer. In the cloudlet layer, the deep learning-based data prediction approach was applied, and in the network layer, data classification by deep learning approach was proposed. Both deep learning approaches use a recurrent neural network (RNN) for classification. In the cloudlet layer, the flow clustering and analysis (FCA) process was employed with the density-based spatial clustring of application with noise (DBSCAN) clustering algorithm. Consequently, using two deep learning algorithms increases the time consumption.

2.3. Research Works on Fog Offloading

Load balancing in fog-based healthcare applications was focused on improving service quality [36]. The fog computing resources were consolidated in this work to balance the load among foglets. Only critical resources were redirected to cloud computing, and the rest requests were scheduled to fog nodes locally. Although this work utilizes fog nodes to prevent overloading, a random selection of fog nodes leads to that the fog nodes to become overloaded.

Multiple agents are deployed in the smart healthcare system to maintain load balancing among fog nodes [37]. The primary aim of the multi-agent fog computing model was to serve critical tasks within the given time. For this, optimal scheduling was enabled by the multiple agents to access the fog resources. Here, the most suitable fog node was selected based on resource availability information. However, numerous criteria are required to process critical and emergency tasks.

A novel four-tier fog architecture was presented to balance load among fog nodes [38]. A dual fuzzy logic algorithm classified the tasks received from the IoT devices to assign priority levels for the tasks. Then the tasks were scheduled based on an artificial neural network. Herein, the single objective is considered for fog node selection, where the energy utility is inefficient.

An energy-efficient offloading strategy was presented for IoT applications [39]. The objective was to achieve better QoS and energy efficiency. For this, a firefly algorithm based offloading scheme was proposed. The firefly algorithm uses a weighted-sum method to determine fitness function. The firefly algorithm is inefficient in local search as it is not suitable for optimal fog node selection.

Comment on Literature: In Table 1, we summarize the literature works with limitations. The analyzes show a massive research gap that needed to be addressed while designing a smart healthcare system.

Table 1. Summarization of literature survey.

Category	Presented Works	Research Issues
Cloud-Smart Healthcare	• PCC and NCC [24] • AHP-VIKOR Model [25] • Fault Monitoring [26] • Statistical Approach [27] • CloudDTH [28] • CNB-Cloud [29]	• Health status detection time is high • Unable to process a large amount of data • Involvement of inefficient algorithms
Fog-Smart Healthcare	• CMI-Fog [30] • Fog-Gateway [31] • STPH [32] • EoT-Hybrid Classifier [33] • HealthFog [34] • UbeHealth [35]	• Low accuracy due to ineffective algorithms • Not able to support a large number of users • High processing time due to the presence of redundant data
Fog Offloading	• Health Load Balancing [36] • Multi-Agent Fog Model [37] • Four-Tier Fog Architecture [38] • QoS Offloading [39]	• Nonoptimal fog selection degrades the offloading performance • single objective is insufficient to select optimum fog node

3. Problem Overview

In this section, we present the problem statement of this work with the background of the problems.

3.1. Overall Problem Statement

In wearable-based smart healthcare, more research works have been presented in the view of a clinical study, and non-clinical home-based health monitoring is concentrated less. The biomedical sensors have been used only for health monitoring, but environmental factors play a pivotal role in healthcare monitoring. Furthermore, fault and duplicate data processing often results in inaccurate outcomes and increases time consumption. Although fog-enabled IoT healthcare systems minimize the latency, this depreciation is not sufficient for emergency events. Because the regional fog node where the number of users is high or the environmental status is worst will become overloaded, it drastically increases the latency for the particular region.

3.2. Background of the Problem

A cognitive dynamics (CDS) concept is presented for smart healthcare and disease diagnosis, along with decision tree-based classification [40]. Here, the decision tree is unstable, i.e., a small change in the data results in a massive change in the structure, which leads to wrong decisions. Further, it is not suitable to handle streaming data like smart healthcare. The overall analysis is performed here in the cloudlet layer, which will increase latency drastically. A structured Gaussian process is proposed for patient-specific physiological monitoring based on its health trajectory [41]. The Gaussian method is a highly parametric approach that considers all parameters known in prior while increasing the complexity. This work finds a single vital sign for determining the health level of the patients.

Furthermore, this work is not suitable to handle mobile users since the cluster formation become complex. Mobile-based physiological sensor system (MoPS) uses a smart shirt and wearable technology [42]. Analysis of the massive data from wearable demands highly efficient algorithms and methodologies. Here the data analysis is performed in a cloud server without using proper algorithms and based on the conventional threshold. It leads to inaccurate classification and also increases the time consumption.

Further, the user status is detected based on the current physiological sign only. But in general, the health status of a user depends upon the time series data. The overall analysis considers a single physiological sign which is insufficient for health status detection.

In a fog-assisted patient health monitoring, Bayesian belief network (BBN) was introduced in the fog layer for health status prediction [43]. In the cloud layer, the temporal information is used to compute the temporal health index (THI) for the patients. In this work, both fog and cloud layers work upon the same attributes (environmental, medical, and behavior). This time consumption increasing makes it not suitable for healthcare monitoring. Processing in the cloud layer increases the latency of emergency events. This work is not appropriate to handle large-scale systems and also unable to support mobile users. However, patients with chronic disease will have mobile nature instead of static. This system is not suitable for real-time analysis. In addition, the BBN classifier is hard to construct and also has low classification accuracy. A hierarchical data fusion method uses the complex event processing (CPE) method for clinical patient monitoring [44]. In each level (sensor, fog, and cloud), data fusion is performed based on a threshold value. All three levels consider biosensor reading only for the patient's health status monitoring. However, environmental factors also play a pivotal role in healthcare monitoring. CPE, which is used in all three levels, uses some pre-defined rules to detect the patient status. However, this is not suitable in the practical scenario since each user has different signs and parameters. Thus, the CPE based analysis is not suitable for real-time analysis. In the fog-cloud smart office healthcare system, the severity index is computed based on environmental, behavior, and posture data [45]. The Bayesian classifier is complex and has less accuracy. Thus, data classification by Bayesian classifiers often results in inaccurate outcomes. The final decision on the user's health status is determined at the cloud layer. It increases the latency considerably for emergency events. Poor load management in the fog layer affects the overall processing ability of the fog layer.

4. Proposed Tri-Fog Health System

In this section, we discuss the proposed tri-fog healthcare system in detail. Each proposed methodology is explained in each subsection.

4.1. System Overview

The proposed tri-foghealth system is designed with three primary layers, such as the wearable layer, the intelligent fog layer, and the cloud layer, as shown in Figure 1. The main intention of the system is to enable remote health monitoring for wellbeing lifestyle. It also supports mobile users. The first layer consists of n users $(u_1, u_2, .; u_n)$ with wearable sensors (we consider a smart shirt with biosensors) and a single PDA device and environmental sensors. The next layer comprises m Data On-Looker nodes $(Ol_1, Ol_2, .; Ol_m)$ and k fog nodes $(F_1, F_2, .; F_k)$. The first layer is responsible for collecting health data from the patients. The second layer involves two different entities, such as data on-lookers and fog nodes. The data on-looker nodes are responsible for redundant data elimination and health status prediction. The health status prediction is the process of the forecasting health status of the user in the next time frame based on the collected time-series data. The result will be added with the current data to increase the accuracy of classification. The fog nodes are responsible for classifying the user's current health status based on the wearable data and predicted report. The final cloud layer consists of a cloud server. The detail of the three layers is provided in the Table 2.

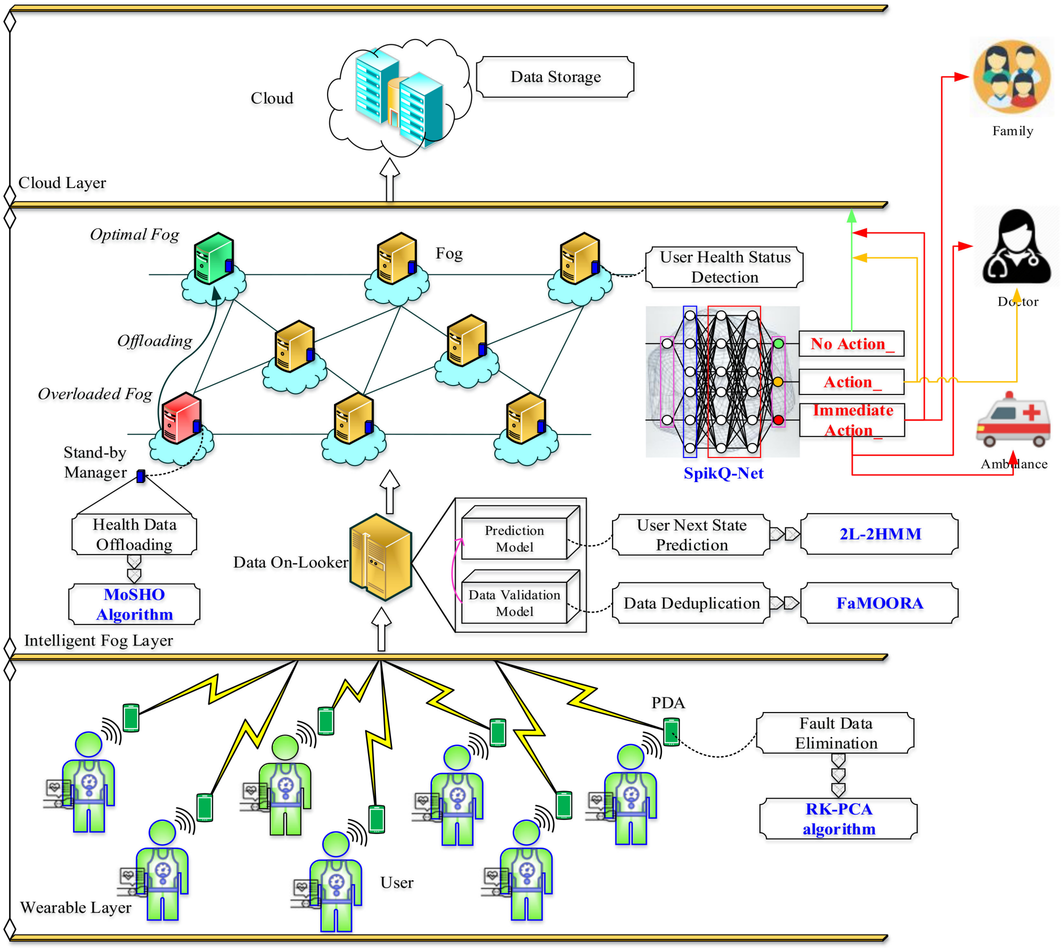


Figure 1. Tri-FogHealth System Architecture.

Table 2. Description of Three Layers in Tri-FogHealth.

Layer	Entities Involved	Functions Incorporated
Wearable Layer	• Users with smart shirts • Wearable Devices with biosensors • PDA	• Data Generation and Aggregation • Fault Data Detection and Elimination by RK-PCA
Intelligent Fog Layer	• Data On-Looker Nodes • Intelligent Fog Nodes	• Redundant Data Elimination by FaMOORA • Health Status Prediction by Temporal Analysis (2L-2HMM) • Health Status Detection by SpikQ-Net • Alert Generation on Emergency Cases
• Cloud Layer	• Cloud Servers	• Data Storage and Retrieval

As shown in the table, each layer works upon different algorithms. Each algorithm is intended for a different process. Each process contributes to improving the quality and accuracy of the healthcare system. On the proposed healthcare system, we make the following assumptions,

- All users have the same number of wearable devices to sense and generate health data;
- The number of fog nodes always equal or higher than the number of data on-looker nodes (i.e.,) $k \geq m$;

By considering the above two assumptions, we designed the tri-foghealth system. Functions carried in each layer of the proposed system are explained in the consecutive subsections.

4.2. Fault Data Elimination in the Wearable Layer

In the device layer, the health data and environmental data are aggregated. The wearable smart shirt is the primary source of health data for each user. Further, the environmental data is collected in every region to detect the user's health status accurately. As the main issue in wearable sensors is fault data generation, the first layer designed in a way such that the fault data problem will be overwhelmed. The fault data is detected and eliminated at the device layer for each user. For that, we present the RK-PCA algorithm in PDA. A fault data in the healthcare system can be detected based on the deviation with the normal data model and the variation with the other sensor data. For instance, the heart rate is generally proportional to the respiratory rate. When there is conflict occurs, then any one of those sensors reported fault data. Thus, the consideration of multiple sensor data models should much support the fault data detection and elimination.

To improve accuracy and processing speed, we present the RK-PCA algorithm. Let u_i wears a smart shirt with w biosensors as $S_1, S_2, \ldots, S_w$. Each biosensor senses the data and reports the data to PDA in a regular time interval. RK-PCA has to find fault data by comparing the current data with the normal data model. To this end, the normal data model is built for the sensor data received from the wearable sensors. The sensor data can be denoted as $\{X = X_1, X_2, .; X_t\}$. The data is captured in d dimensions in t time instances. The data model is represented as the Eigenvector. The Eigenvalue equation is defined as $\sum_{\varphi} e_i = \alpha_i e_i$. In this, e_i is estimated as;

$$e_i = \sum_{j=1}^{t} \alpha_i(j)\varphi(X_i(z)) \tag{1}$$

Here, X_i is the chosen data from the normal data model. Similarly, the eigenvalue is computed for all normal data values. From the normal data model, the fault data is detected, as shown in Figure 2.

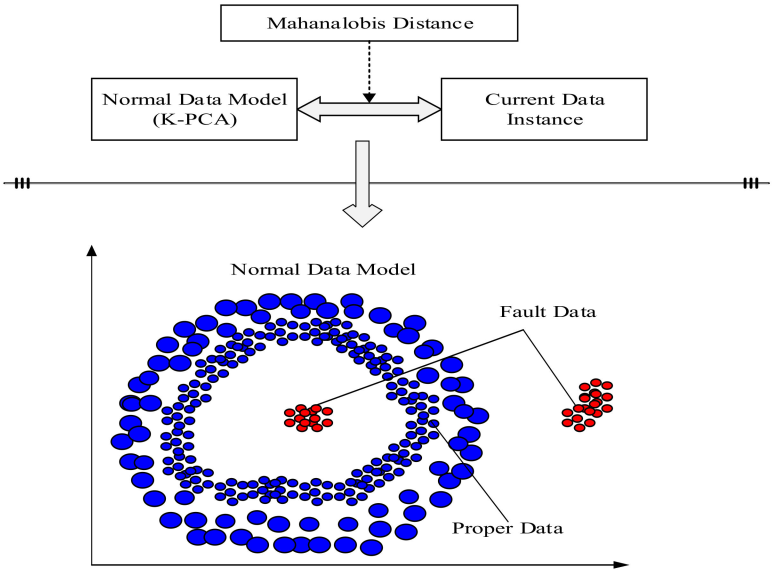


Figure 2. Fault Data Detection by Rapid Kernel Principal Component Analysis (RK-PCA).

As in the figure, the first step is to model the normal data by the RK-PCA algorithm. Here, kernel selection is carried as follows,

$$Ker(X_i, X_j) = \exp\left(\frac{1}{2\sigma^2}(X_i - X_j)^T Q^{-1}(X_i - X_j)\right) \quad (2)$$

The considered kernel is Mahalanobis Kernel (MK) since it supports the non-linear transformations of the data points.

The next step is to compute the similarity between current data instance (S_C) and normal data model (S_{nor}). The similarity is computed as Mahalanobis distance as follows,

$$dis(S_c, S_{nor}) = \sqrt{(S_C - S_{nor})^T CM^{-1}(S_C - S_{nor})} \quad (3)$$

Here, CM defines the covariance matrix defined by the eigenvalues of the normal data model. If the distance is low, then the data is normal. Otherwise, the data is a fault. The PDA carries this process, and the fault data is suppressed in the PDA. In Algorithm 1, we provided a detailed algorithm for RK-PCA algorithm-based fault data detection. The information with no fault is then transmitted to the next layer (i.e.,) intelligent fog layer.

Algorithm 1: RK-PCA for Fault Data Detection.

```
Initialize $X_i$
For all $u \in Layer$ 1
    Get $\{X_i\}$ from $u_i$
    Set RK-PCA Dimension
    Select $Ker$ function          //Equation (2)
    Build $e_i$                    //Normal Data Model
    Construct $CM$                 //From $e_i$
    Compute $(dis)$                //Mahanalobis Function
    If $X_i$ $to$ $Fault$
        Drop $X_i$
    Else
        Send $X_i$ $to$ $Layer$ 2

    End If
End For
End
```

4.3. Redundant Data Elimination by Data Validation

In the fog layer, two primary functions are performed by the data on-looker nodes. The first process is redundant data elimination, and the next process is the user's health status prediction. The faultless data from PDA is received by the data on-looker nodes in the region. The data on-looker nodes are the special nodes that maintain the previous data records from the wearable devices and environmental sensors. The data on-looker nodes are designed with data validation model and prediction model. The data validation model is responsible for redundant data elimination by comparing the current data with previous data instances. Then, the redundant free data is fed into the prediction model to perform first-level health status prediction. The conceptual structure of the data on-looker node is illustrated in Figure 3.

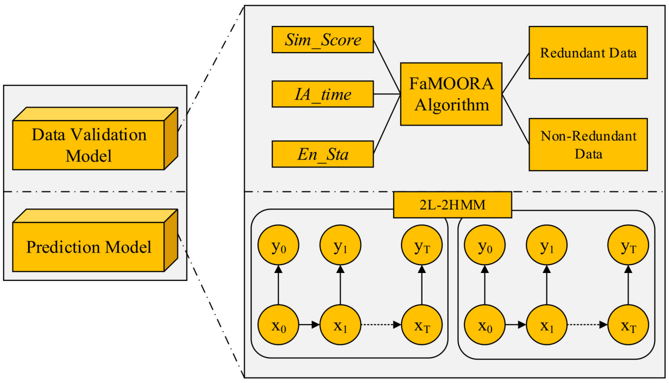


Figure 3. Structure of Data On-Looker Node.

4.3.1. Redundant Data Elimination

In this work, we intend to eliminate redundant data without affecting emergency data. In general, the redundant data elimination is performed if any one of the instances in prior records is matched with the current data. However, in the healthcare system, a redundant may also be emergency data. In this case, the data analysis will be affected. Thus, the problem of redundant data elimination is formulated as a decision-making problem and resolved by a hybrid multi-criteria decision-making algorithm.

We present the FaMOORA algorithm that works upon three major criteria, such as similarity score (Sim_Score), inter arrival time (IA_Time) and environmental status (En_Sta). Each criterion can be defined as follows,

Sim_Score—The current data instance D1 is verified with prior data records from the same user. D1 is the current data instance from the ith user. A data instance is made up of sensor readings sensed by the biosensors incorporated on the user. Thus, D1={Sd1, Sd2, ; Sdw}. Then D1 is compared with succeeding data D2, D3, . . . , Dq of the same sensor of the same user:

$$Sim\_Score\left(D_1, D_j\right) = \sqrt{(x_1 - x_2)^2 + \left(y_1 - y_j\right)^2} \tag{4}$$

The score is computed between all prior data instances.

IA_Time—The inter-arrival time is considered to find whether D_1 is needed to be eliminated or not. If the time between two similar data is relatively less, then the data is duplicate. If the time difference is high, then the data is new, and it is essential for data analysis.

En_Sta—The difference in environmental status also plays a vital role in redundant data elimination. It is because the biosensor data may be the same, but the environmental data may be different. In the worst environmental scenario, the biosensor data will be necessary. In this scenario, data must be preserved.

All three criteria are considered in the FaMOORA algorithm for making decisions on the current data instance. The procedure of the FaMOORA algorithm is detailed below.

Firstly, the decision matrix (DM) is constructed with data upon three criteria as follows,

$$DM = \begin{bmatrix} SS(1,2) & SS(1,3) & SS(1,q) \\ IA(1,2) & IA(1,3) & IA(1,q) \\ EN(1,2) & EN(1,3) & EN(1,q) \end{bmatrix} \tag{5}$$

Then DM is normalized, taking the square root of the sum of all the squares of each alternative per attribute. This ratio can be given as,

$$SS(1,q)^* = \frac{SS(1,q)}{\sqrt{\left[\sum_{q=2}^{q} SS(1,q)^*\right]}} \tag{6}$$

From the normalized DM, the attributes are weighted by using the fuzzy approach. In conventional MOORA [46], the weight value is given for the characteristics based on the analytic hierarchy process (AHP). However, it consumes a significant amount of time. Thus, we present a fuzzy weighting scheme for the FaMOORA algorithm. The weight value of the attribute *SS* is given as,

$$nSS_q = \sum_{q=1}^{g} w_q SS(1,q)^* - \sum_{q=g+1}^{h} w_q SS(1,q)^* \tag{7}$$

Here, g denotes the number of attributes that need to be maximized (beneficial attributes), and h indicates the number of attributes to be minimized (non-beneficial attributes). Further, $\mathrm{nSS_q}$ is the normalized attribute value. Then the fuzzy weight value is computed within the range of [0, 1]. The weight value calculated above will be in [$\mathrm{r}_{1,}$ r_2]. Then it can be mapped into fuzzy values as,

$$FW\left(nSS_q\right) = \frac{\left|nSS_q - r_1\right|}{|r_1 - r_2|} \tag{8}$$

For each attribute value, the weight value is computed as above. Thus, the normalized DM becomes the fuzzy weighted DM (FWDM) with the values of [0, 1]. From the FWDM, the decision on the current data instance is made based on the following if-then rule,

$$\text{If } (\text{SS}(1,\text{q}) \text{ to } (>0.5)\ \&\text{IA}(1,\text{q})\&\&\text{EN}(1,\text{q}) \text{ to } (<0.5)$$
$$\text{Then, } \text{D}_1 = \text{Duplicate}$$

By following the above rule, the data duplication decision is made, and the data on-looker nodes suppress the duplicate data. Then, the redundant free data is fed into the nearby fog node for health status prediction.

4.3.2. User Health Status Prediction by Temporal Analysis

The prediction model receives the user's health data from the data on-looker node then analysis the data based on temporal information. The prediction module carries the process of health status prediction. In this section, we introduce the 2L-2HMM model that predicts the users' health status. In this stage, the data instance from biosensors is analyzed with time series. The proposed 2L-2HMM works upon two levels. In the first level, the biosensor data is interpreted as in the HMM model. Let D_i be the data instance of u_i. The fog node first extracts the time series data u_i of the u_i. The next status of u_i upon biosensor data is characterized by $\delta_i(t+1)\ of\ \{\pi, \mathbb{A}, \mathbb{B}\}$. Here, *A* denotes the transition probability matrix, and *B* is the observation probability matrix. The observation matrix is composed of the data instances gathered by the biosensors. The initial state distribution is denoted as,

$$\pi_i = P(H_1 = 1), \qquad 1 \leq i \geq p \tag{9}$$

where, *P* is the number of probability distributions, and *H* denotes the hidden state. The state transition probability of transition from state *x* to state *y* is computed as,

$$\mathbb{a}_{x,y} = P(H_t = y|H_{t-1}), \quad 1 \leq y \geq p,\ 1 \leq k \geq w \tag{10}$$

The observation probability distribution defines the probability of state transition for the observation lk as,

$$\mathbb{b}_k = P(l_k|H_t = y), \quad 1 \leq y \geq p,\ 1 \leq k \geq w \tag{11}$$

In general, the observation variable lk depends on the hidden state H_k while the hidden state depends on the previous hidden state H_{t-1}. The prediction of the next state of u_i is performed based on the Bayes assumption on observation distribution as follows,

$$P(l_k|H_t = y) = \prod_{w=1}^{|L_k|} P(l_k|H_t = y) \tag{12}$$

The user's health status is determined as normal or emergency based on the time-series data.

In the second level, the location variations of the users are considered. For instance, the set of locations of the user denotes $= \{L_1, L_2, \ldots, L_t\}$. As the tri-foghealth works upon remote monitoring, the user location may cover home, home surroundings, home long-distance. In this level, the user's current location is verified, whether it is frequent or non-frequent. Upon this information and the status predicted from the first level, 2L-2HMM uses the following rules,

- When the user's next status is emergency, and the location is not-frequent of the user, then the user is in a critical situation (C_1).
- When the user's next status is emergency, and the location is a frequent location of the user, then the user is in a near-critical situation (C_2).
- When the user's next status is normal, then the user is in the non-critical situation (C_3).

The user's next health status is appended with the data to increase the accuracy in the fog nodes. In addition, the health status prediction process assists in timely data analysis. In Algorithm 2, we detailed the process carried by data on-looker nodes, including redundant data elimination and user's health status prediction. At the end of this algorithm, the data instance is appended with the user's next state to make health status detection accurate.

Algorithm 2: Process in Data On-Looker.

```
Initialize   $D_1$, $D_2$, ..., $D_j$, ...
//Start Validation Model→ FaMOORA
For   $D_i$ do
    Compute *Sim_Score*
    Compute *IA_Time*
    Compute *EN_Sta*
    Construct *DM*
    Normalize *DM*
    Update weights by *FW*
    Apply IF-THEN Rules
    If *Di* to Duplicate
        Drop $D_i$
    Else
        *Fed   $D_i$ to 2L – 2HMM*
    End if
End for
//Start Prediction Model  →2L-2HMM
For Di do
    Extract  $\mu_i$
    Compute  $a_{x,y}$, $b_k$                //Equations (10) and (11)
    Predict Next State
    Extract  $\theta = \{L_1, L_2, \ldots, L_t\}$
    Fuse Two Levels
    Append User's Next Status
End for
End
```

4.3.3. User's Health Status Detection by Fog Nodes

After predicting the user's health status by data on-looker, the health data is delivered to the fog nodes. In fog nodes, health status detection is performed based on three categories of attributes by using SpikQ-Net. The proposed SpikQ-Net combines a spiking neural network that mimics the natural neural network [47] and a quantum neural network that works upon quantum mechanics [48]. The fusion of both neural networks results in better accuracy of the user's health status detection process. First, we clearly described the attributes detail in the Table 3.

Table 3. Attributes Description.

Category	Attributes	Description
Behavior	• Cholesterol • Smoke • Cig/Day • Height • Weight • Activity • Chronic_Disease • Location	Attributes are related to the user's normal behavior. We consider the static information like smoke or not and dynamic information like location to monitor the user's health status
Biomedical	• Blood_Pressure • Glucose_Level • Heart_Rate • Respiratory_Rate • Temperature	We consider the data from wearable devices. All data related to the biomedical data that shows the user's current health status
Environmental	• Room_Temperature • Noise_Level • Air_Quality	These data are collected from the regional environmental sensors.

The SpikQ-Net gets the input of these three categories of attributes, such as behavioral, biomedical, and environmental. Generally, the user's health status is detected based on the biomedical data. The reason for considering behavioral and environmental attributes is that both attributes have a high impact on the biomedical sensor data. For example, the heart rate is increased when the user is doing exercise and also in a noisy environment. In this case, the abnormal heart rate is reported by the biomedical sensors due to the tenancy of environmental and behavioral attributes. Thus, we considered all three categories of attributes in the user's health status detection. All three types of attributes are fed into the SpikQ-Net from the current data instance, as shown in Figure 4.

Let the current data instance be $D_i(t)=\{(BD_i)(BMD_i)(EN_i)\}$. The data instance at time t is composed of behavior (BD_i) biomedical (BM_i), and environmental (EN_i) attributes at time instance t. For each user, SpikQ-Net first extracts these attributes by learning the data received. Then, the weight value is adjusted based on the fusion of these attributes to detect health status. Similarly, the instances from various users are fed as input to the SpikQ-Net. The proposed SpikQ-Networks as follows,

1. The set of input functions $(D_1(t),\ D_2(t), .; D_i(t), \ldots, D_n(t))$ is fed into the input layer;
2. Each neuron learns the input to produce output vectors in the hidden layers;
3. The initial weight values (W_i) are initialized as the random numbers;
4. Then the weight values are updated according to the update rule of quantum mechanics;

$$W_i(t+1) = W_i(t) + \eta(\,|\,o\rangle - |y(t)\rangle)\langle D_j\,| \tag{13}$$

5. Here o is the desired output provided for the training stage, η is step size, and it is set as $0 < \eta < 1$;
6. At last, the out layer learns the output based on the weights learned from hidden layers. The output of the quantum perceptron at the time t is,

$$|y(t)\rangle = \sum_{j=1}^{n} W_i(t)|D_i\rangle \tag{14}$$

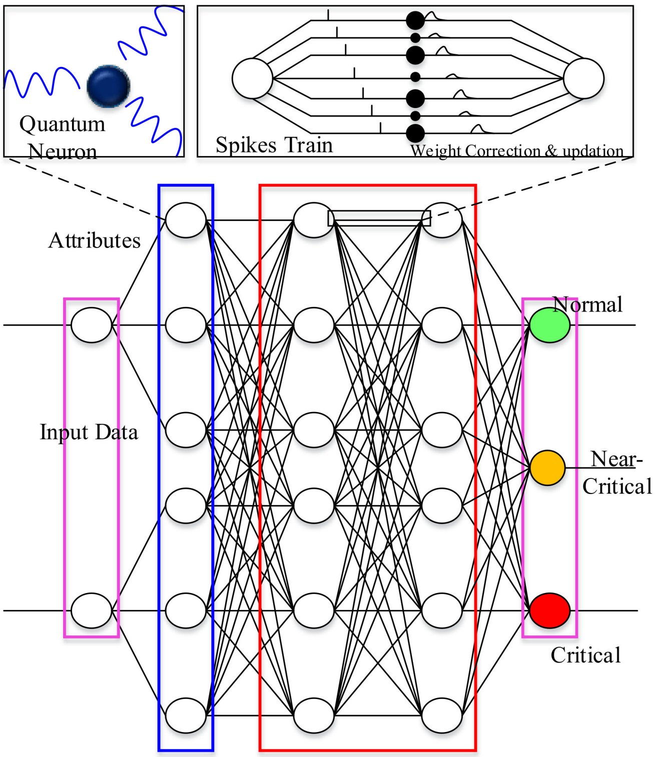


Figure 4. SpikQ-Network for Health Status Detection.

In this procedure, we introduce spikes between quantum neurons. The current activation function value is considered the neuron's state, and the spikes increase its value to achieve better accuracy. The introduction of spikes alters the weight update expression as follows,

$$W_i(t+1) = W_i(t) + \eta\delta_i(t) + v\Delta W_i(t-1) \tag{15}$$

Here, v is the momentum parameter and $\Delta W_i(t-1)$ is the weight correction in the previous iteration. In this manner, the weight values of each neuron are updated for each parameter, and the output layer detects the user's health status.

The main aim of this process is to detect the user's health status as normal, critical, and non-critical classes. As per the class, the necessary action is taken. The fog nodes take all the steps. That is, the latency is relatively low for the users. In addition, emergency actions assist in the timely detection of abnormalities as the process is performed in the fog layer.

4.3.4. Health Data Offloading

The data on-looker node assigns the data for analysis to random fog nodes. Thus, the fog nodes become overloading. To overcome this issue, we present the health off procedure. Although data processing in the fog layer minimizes latency, there has been a huge issue that is fog overloading. Users within the same region often access the same fog node. In such a case, when user density is high in a particular area, the fog node will become overloaded. Besides, if the environmental condition of a specific region becomes worst, then many users in that region will require immediate action. Thus, we present a fog offloading scheme in healthcare services. Each fog node has the stand-by manager who monitors the load of the fog node regularly and makes a decision on fog overloading. The health off decision is made in the following two cases,

Case 1: When the fog node becomes overloaded

Case 2: When the environmental condition becomes worst

If the health off decision is made, the stand-by manager selects an optimal fog node for health data offloading. For optimal fog node selection, we propose the MoSHO algorithm. Consider, f_{ol} becomes overloaded, then it selects the optimal fog node (fopt) from the set of fog nodes. The spotted hyena optimization algorithm is a bio-inspired optimization algorithm that works upon the social behavior of spotted hyenas [49]. As single objective-based offloading is inefficient to maintain the load among fog nodes, we formulate multiple objective functions in the MoSHO algorithm. The proposed MoSHO algorithm works in three steps: searching prey, encircling, and attacking the target. The overall process of the MoSHO algorithm is summarized below.

Fitness Evaluation—Initialize the spotted hyenas' population. The initial population list consists of a set of candidate fog nodes as $IP \in \{F_1, F_2, .; F_k\}$. Then, the fitness value is computed for each node in the IP set. In the MoSHO algorithm, multiple criteria are considered for the computation of the fitness function of j^{th} candidate search agent $Fn(F_j)$. The considered multiple criteria are load value β, distance τ, and energy required ρ. The first objective function computes the current load value of the candidate fog nodes in terms of the number of processing tasks and available resources. The second objective function computes the distance between the current and candidate fog nodes. Thus, the offloading is performed between two nearby fog nodes, which minimizes the offloading time. The third objective function is formulated based on the energy required for offloading the current load from the current fog node to the optimal fog node. Thus, the fitness value is expressed as follows,

$$Fn(F_j) = \frac{1}{\sum(\beta_j, \tau_j, \rho_j)} \tag{16}$$

The fitness function is computed for all fog nodes in the candidate set.

Hunting—From the candidate solutions, the search agent who has the highest fitness value is selected to explore solutions in the given search space. The best search agent has the location of the prey. Thus, the other search agents form a cluster to move towards the best search agent. The cluster and the movement are expressed mathematically as follows,

$$D_h = \left|B.\omega_{h_1} - \omega_{h_2}\right| \tag{17}$$

$$\omega_{h_2} = \omega_{h_1} - E.D_h \tag{18}$$

$$C_{h1} = \omega_{h_2} + \omega_{h_2+1} + \ldots + \omega_{h_2+V} \tag{19}$$

Here, ω_{h_2} defines the position of the first best-spotted hyena, ω_{h_1} represents the position of other spotted hyenas. In addition, V denotes the number of spotted hyenas, and it is computed as follows,

$$V = Count_{nos}\left(\omega_{h_2} + \omega_{h_2+1} + \ldots + (\omega_{h2} + Ran)\right) \tag{20}$$

where Ran defines the random vector [0.5, 1], nos represents several solutions and count all candidate solutions.

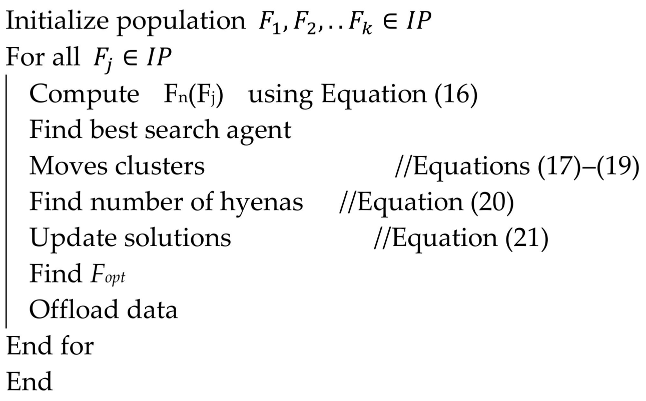
Attacking Prey (Exploitation)—In this phase, the solutions are updated towards the best search agent. This phase is known as attacking the prey. It can be mathematically formulated as,

$$\omega(b+1) = \frac{C_h}{V} \tag{21}$$

The $\omega(b+1)$ saves the best solution and updates the positions of other search agents. The position updation is performed according to the best search agent. Next, the search agent fitness value is updated towards the ideal solution. This process is iterated until the optimal fog node is selected.

In Algorithm 3, the procedure of MoSHO based health off is proposed. Then, the data is offloaded to the optimal fog node selected by the MoSHO algorithm. The process of emptying prevents the fog nodes from overloading the problem. As offloading is performed between closest fog nodes, the latency is minimized. All three functions are performed in the fog layer to achieve better performance.

Algorithm 3: Procedure of Health Off by MoSHO.



```
Initialize population $F_1, F_2, .. F_k \in IP$
For all $F_j \in IP$
    Compute   $F_n(F_j)$   using Equation (16)
    Find best search agent
    Moves clusters                //Equations (17)–(19)
    Find number of hyenas     //Equation (20)
    Update solutions              //Equation (21)
    Find $F_{opt}$
    Offload data
End for
End
```

5. Experimental Evaluation

In this section, we present the experimental setup and comparative analysis of the proposed smart healthcare system.

5.1. Experimental Setup

In this section, we illustrate the experimental model of the tri-foghealth system. The proposed tri-foghealth is modelled in the iFogSim simulation tool. The iFogSim tool is best to model fog based environments with the functionalities of CoreCloudSim on the top of the cloud layer. For the proposed system, the code is written in Java language. All necessary packages and tools are installed on the PC to make experiments on the proposed tri-foghealth system. The proper software and hardware parameters are provided in the Table 4.

Table 4. Software and Hardware Description.

Software/Hardware	Description
Simulation Tool	iFogSim
Simulator Version	3.0.3
Operating System	Windows 7 Ultimate
Programming Language	Java
Development Kit	JDK 1.8
IDE	NetBeans 8.2
Database	MySQL-5.1.36 (Wamp Server 2.0)

The MySQL database is used to store the healthcare data of users. The other simulation parameters set in the experimentation is provided in the Table 5.

The proposed tri-foghealth system is built up based on the above simulation settings. At first, the users are needed to register with the proposed method by reporting their behaviors in the initial form. Then, the user data is stored in the cloud layer. Data from the same user is stored in a separate database after analysis by the fog nodes. The proposed experimental setup and flow are illustrated in Figure 5. As in Figure 5, we construct our fog topology for the smart healthcare system. The users and the fog nodes are deployed in different regions. The proposed tri-foghealth system analyzes the data from another region.

Table 5. Experimental Settings.

Parameter		Value
Number of users		≅ 50
Number of biosensors in each user		5
Number of data on-lookers		10
Number of fog nodes		10
Fog Node Configuration	Storage	1 GB
	Bandwidth	1000 KBs
	Resource Cost	3.0
	Memory Cost	0.5
Cloud Configuration	MIPS	1000
	Memory	10 MB
	Bandwidth	1000 KBs
SpikQ-Net	Number of neurons	10
	Number of hidden layers	3
	Activation Function	Sigmoid
	Step Size	0.2
MoSHO	Initial Population	100
	Ran	0.3
	Iteration	100

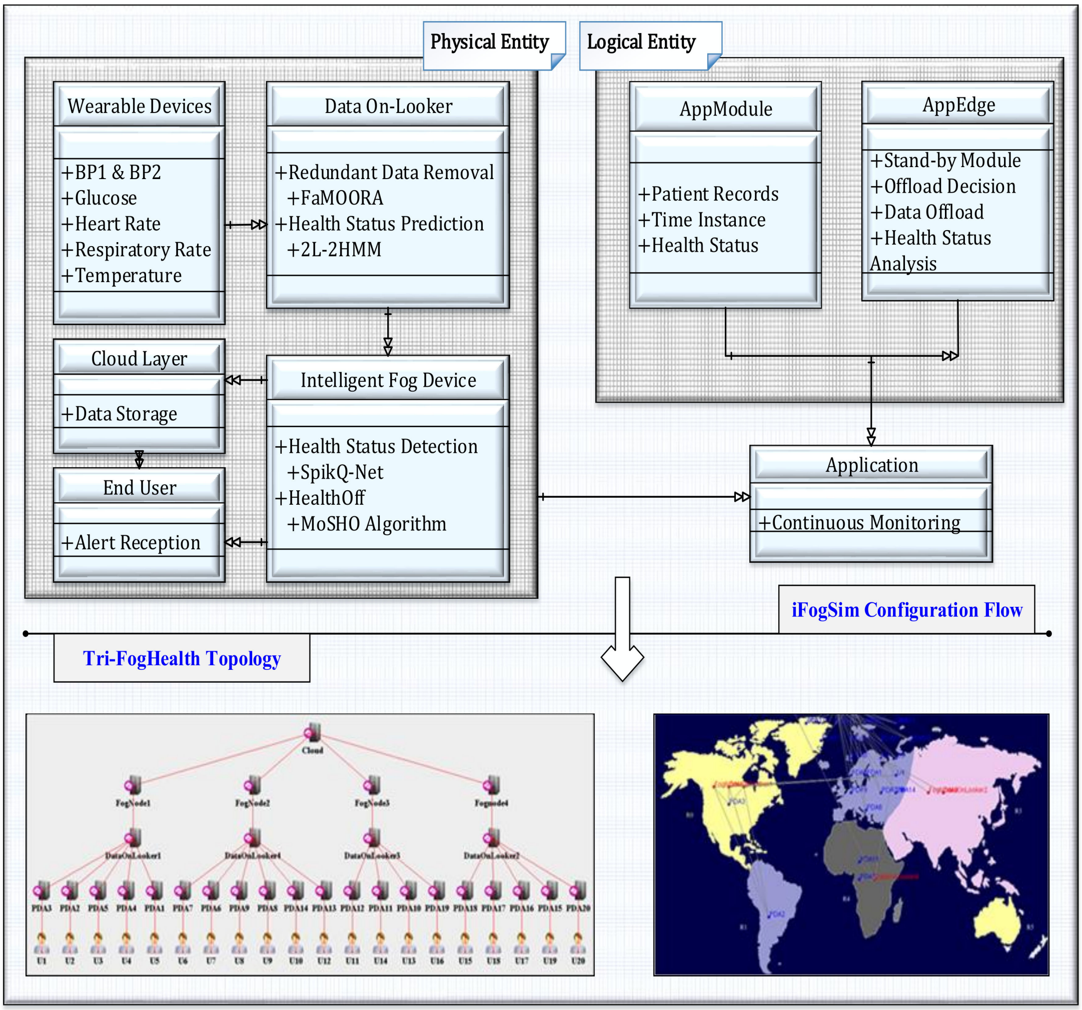


Figure 5. Experimental Setup of Tri-FogHealth System.

Data generation and analysis: in this work, we present a framework for monitoring the user's regular health status. Thus, we generate health data from the sensors. For each user, we create health data as per the considered attributes. The sample data generated for the users in a given time instance in a synchronized manner, as shown in the Figure 6b. In this way, we create nearly 1000 sample data with biomedical, behavior, and environmental attributes (i.e.,) each sample represents the biomedical, behavior and ecological data. The data is generated from 50 users registered in the system. The generated samples are sufficient to train the SpikeQ-Net neural network. Then, we use this data to train our healthcare system. Figure 6c shows the data points classified by the 2L-2HMM algorithm. Then, these data points are fed into SpikQ-Net, which is shown in the Figure 6d. Here, we set the binary values for specific attributes. For instance, we set Activities as Sit_(0), Sleep_(1), and Work_(2). Thus, the data contains either 0 or 1 or 2 at a given time instance. The proposed system also supports a particular disease diagnosis if the proper dataset is used.

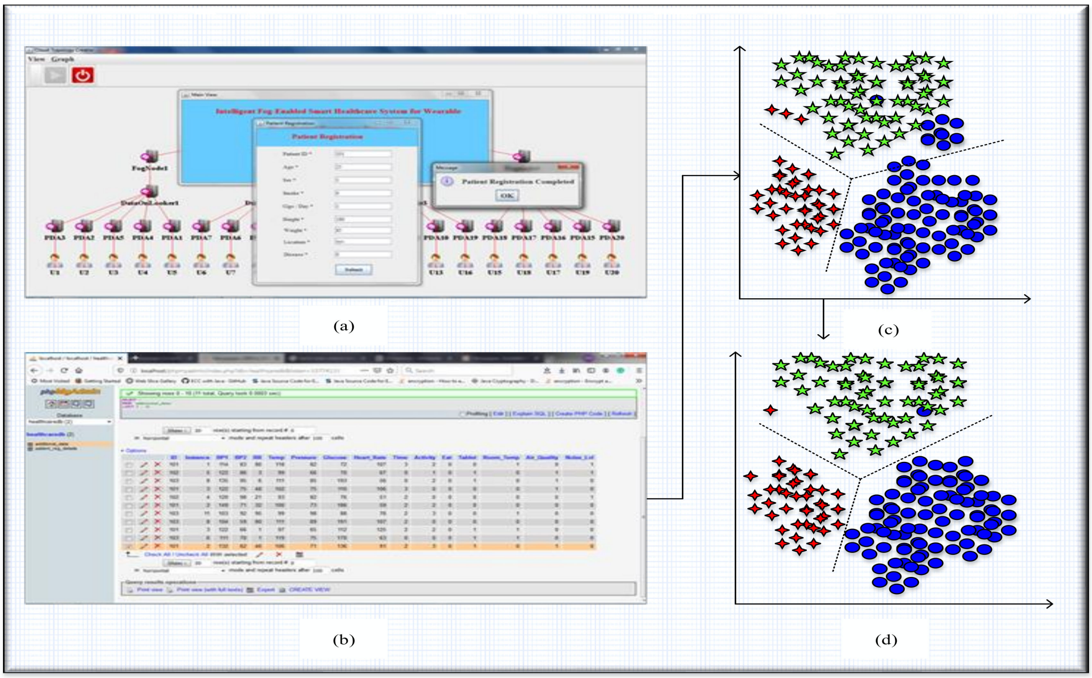

View Graph
Intelligent Fog Enabled Smart Healthcare System for Wearable
Patient Registration
Patient Registration Completed
OK
FogNode1
DataOnLooker1
PDA3 PDA2 PDA5 PDA4 PDA1 PDA7 PDA6
PDA10 PDA19 PDA18 PDA17 PDA16 PDA15 PDA20
U1 U2 U3 U4 U5 U6 U7
U13 U16 U15 U18 U17 U19 U20
(a)
ID Instance BP1 BP2 RR Temp Pressure Glucose Heart_Rate Time Activity Eat Tablet Room_Temp Air_Quality Noise_Lvl
(b)
(c)
(d)

Figure 6. (**a**) User Registration in Tri-FogHealth, (**b**) Sample Data Generated, (**c**) Health Status Prediction (2L-2HMM), (**d**) Health Status Detection (SpikQ-Net).

5.2. Application Scenario: Remote Quarantine Monitoring for Covid-19 Outbreak

The novel corona virus (Covid-19) outbreaks are the significant and severe outbreak that healthcare systems handle. In Covid-19, it is impossible to treat all patients in the hospitals since the count is increasing rapidly. In such cases, remote patient monitoring will be a practical solution [46]. It can be achieved by our proposed tri-fog heath system, which focuses on remote monitoring. In Covid-19, most of the affected peoples have mild external symptoms like fever, cough and throat pain but have severe internal symptoms such as oxygen level reduction. As a result of these symptoms, the patient suffers from respiratory problems. We can apply the proposed tri-fog health for monitoring the patients who are in self-quarantine. The application setup is demonstrated in Figure 7.

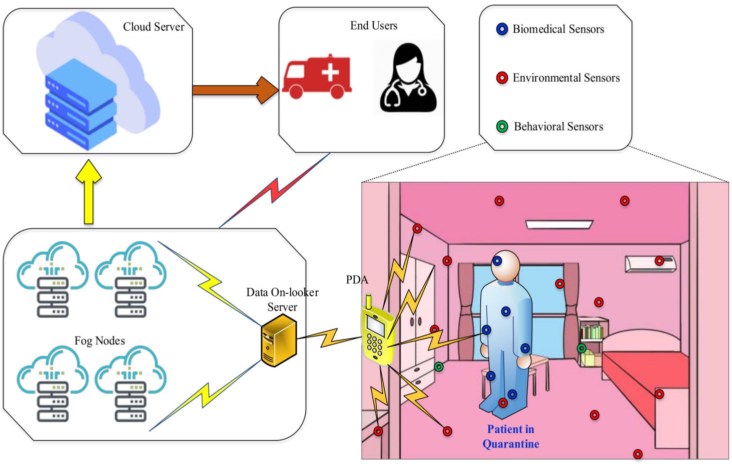


Figure 7. Covid-19 Remote Monitoring by Tri-FogHealth.

As shown in the setup, the patient who is quarantined in a room is equipped with biomedical sensors such as BP, glucose, respiratory rate, and temperature. Additionally, the place is equipped with environmental sensors to measure environmental changes. Before quarantine, the patient must be registered in the system by providing health history, which is further helpful in analyzing health status. The data collected from the quarantine room is sent to the data on-looker node through wireless communication. There, the data is cleaned, and the health status is predicted. Then, the information is fed into the fog nodes, which detect the health status by SpikQ-Net upon all three sets of attributes. It is worth to mention that tri-fog health generates an immediate alert to hospitals and doctors if any abnormality is detected in data. Corresponding government authorities can handle the fog and cloud servers. Thus, patient monitoring will be easy and accurate. Above all, the patients will get timely health services, which prevent early deaths due to Covid-19. In addition, the involvement of the health off procedure assists in offloading data among regions if any one of the regions become more congested. This prototype can be implemented by deploying wearable devices for quarantined patients. Nowadays, the usage of wearable (smart watches) becomes interesting among many users same as smartphones (can be used as PDA). Thus, data collection and analysis is easy in the proposed prototype.

5.3. Comparative Analysis

After setting up all necessary parameters, we generate data in the system to compare the efficiency of the proposed method. The proposed tri-foghealth system is compared with prior healthcare systems. For comparison, we consider fog based healthcare systems such as Fog-BBN [43], Fog-CPE [44], Fog-Smart Office [45] and also cloud-based healthcare systems such as Cloud-CDS [40], Cloud-Gaussian [41], and Cloud-MoPS [42]. We have summarized comparison between all these works in the Table 6. In the considered results, attributes are also varying. Fog-BBN and Fog-Smart Office works have considered all three sets of attributes, including biomedical, behavior, and environmental attributes. The rest of the current research works consider a single set of attributes (i.e.,) biomedical attributes. Besides, none of the existing healthcare works has concentrated on the offloading procedure in fog computing. Thus, the health load among fog nodes is imbalanced, which affects the overall performance.

Table 6. Comparison of Prior Works.

Work	Architecture	Concept	Attributes Considered	Cons	Common Issues
Cloud-CDS [40]	Cloud-based Smart Healthcare System	CDS based Healthcare System Decision Tree-based classification	Biomedical	1 The involvement of decision tree increases latency and also it is not stable 2 The CDS system is unable to handle streaming data for healthcare services	1 Latency is high since the data processing is performed in the cloud layer
Cloud-Gaussian [41]	Cloud-based Physiological Sign Monitoring	Gaussian Model detects health level Patient's Health Trajectory is extracted	Biomedical	1 The high Gaussian parametric approach increases the complexity 2 Single physiological sign consideration limits the accuracy of abnormality detection 3 Not able to support mobile users	2 Processing redundant data in the cloud layer improves execution too 3 Not suitable for emergency services
Cloud-MoPS [42]	Cloud-based Mobile Healthcare System	A smart shirt and Wearable Technology is used for health monitoring	Biomedical	1 Massive data analysis by threshold technique is inefficient 2 Classification accuracy is low, but the latency is high	
Fog-BBN [43]	Cloud-Fog based Patient Health Monitoring	BBN Classifier is used in Fog Layer THI is computed in Cloud Layer	Biomedical Behavior Environmental	1 Processing the same attributes in fog & cloud layer increases latency 2 BBN classifier is hard to complex and analyze 3 Only supports static users	1 Imbalanced load among fog nodes affects the system stability 2 Execution time is high due to the presence of a fault and redundant data from wearable devices
Fog-CPE [44]	Hierarchical Data Fusion Model with Sensor, Fog and Cloud Model	Biosensor Readings are Classified in each Level	Biomedical	Biosensor data alone is insufficient to analyze the health data accurately CPE works upon pre-defined rules which are not efficient for dynamic health data	
Fog-Smart Office [45]	Fog-Cloud Healthcare System for Smart Office	Severity Index is constructed to monitor the health status	Biomedical Behavior Environmental	Bayesian classifier which is used here is complex and has less accuracy Final health status is detected and a decision made in the cloud layer which increases latency	

The proposed tri-foghealth system resolves the major issues encountered in the prior research works. Thus, the proposed work achieves better results. The comparisons are made based on latency, execution time, detection accuracy, and system stability.

5.3.1. Analysis of Latency

Latency is defined as the overall time taken by the system to respond to the user's received data. The latency includes propagation and processing time. Latency (α) is computed as follows,

$$\alpha = \alpha_p + \beta \tag{22}$$

Here, α_P denotes the propagation time taken by the data to reach the fog layer and β indicates execution time (i.e.,) processing time. In Figure 8, we compare the latency achieved by proposed and prior works concerning the number of users.

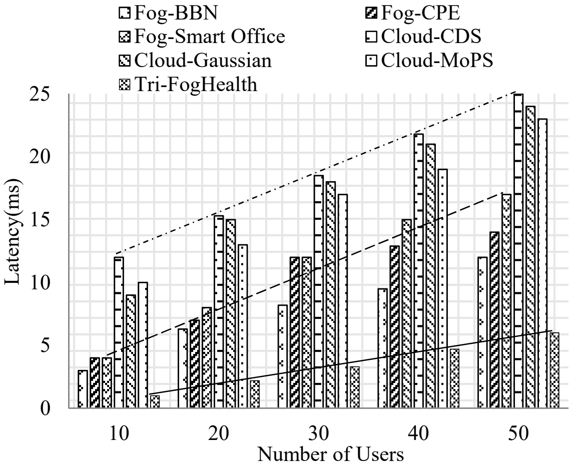


Figure 8. 7 Comparison on Latency upon Number of Users.

The analysis shows that the proposed tri-foghealth system has minimum latency than other cloud-based and fog-based systems. The cloud-based approaches like CDS, Gaussian, and MoPS methods have latency up to 20 ms, which is twice the time higher than the fog-based systems. Among the fog based systems, the proposed Tri-FogHealth system attains latency 6 ms even for 50 users. In the CDS system, a decision tree is used for classification. In general, the decision tree has higher time consumption, which increases the latency for users.

Similarly, the Gaussian approach has high complexity, and the MoPS algorithm works upon a static threshold, which can't handle the massive amount of data. Besides, the cloud-based systems relatively introduce high latency for users. Although BBN, Smart Office, and CPE methods decrease latency better than the cloud systems, still latency is high due to inefficient algorithms.

In Figure 9, we compare the latency achieved by fog-based approaches based on the number of fog nodes. Latency in fog-BBN is high since the second level THI computation is performed in the cloud layer. Thus, the latency is high in fog-BBN. In fog-CPE, rule-based health status detection increases the latency in the presence of a small number of fog nodes. In the fog-smart office, the Bayesian classifier has higher time consumption. Each fog-based approach has some limitations, such as increasing the latency and decreasing the efficiency of fog nodes. In proposed tri-foghealth, the system minimizes the latency with the efficient algorithm. The tri-foghealth system has only 5 ms of latency, even with two fog nodes. The main reason behind this high efficiency is that the tri-foghealth system follows the health off procedure in cases of fog overloading. Therefore, the issue of fog overloading has been overwhelmed. Thus, the proposed tri-foghealth system achieves much better efficiency.

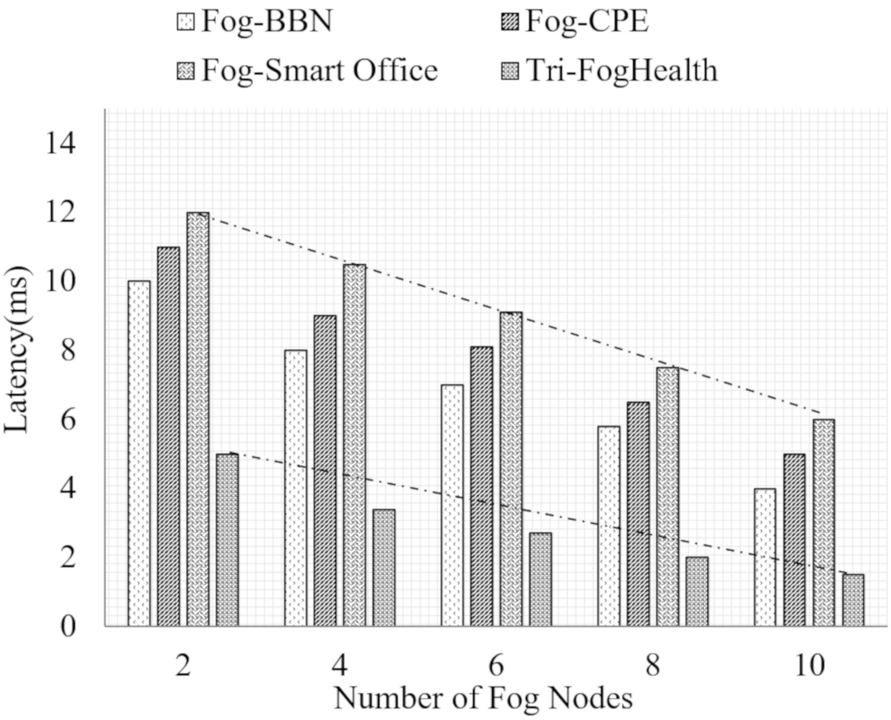


Figure 9. Comparison of Latency upon Number of Fog Nodes.

5.3.2. Analysis of Execution Time

Execution time is defined as the time taken by the healthcare system to predict the users' health status. The execution time measures the health status prediction and detection time.

In Figure 10, we compare the execution time achieved by the proposed tri-foghealth system. And the analysis shows that the execution time performed by the proposed tri-health system is lower. In comparison, the main reason for high execution time is the inefficiency of the algorithm. In fog-BBN, the Bayesian classification and THI computation are performed based on the same attributes (i.e.,) processing the same attributes increases the execution time. In the fog-CPE and smart office, the conventional rule-based algorithms are incorporated. Thus, the execution time is high in all these works. To mention that the Cloud-MoPS method takes 19 ms to process data from 50 users. As MoPS incorporates no algorithm for data analysis, it takes much time, which is nearly thrice time higher than the tri-foghealth system. In Figure 11, we analyzed execution time based on the number of fog nodes.

The comparative analysis shows that all of the fog based approaches suffer from higher execution time. Another leading cause for excessive execution time is that all works analyze fault and duplicate data from wearable devices. This analysis increases the execution time for the user's health data up to 15 ms. In addition, the curve shows that fog-smart office work has 8 ms of execution time, even in 10 fog nodes. The increase in the number of fog nodes alone is ineffective in achieving better efficiency.

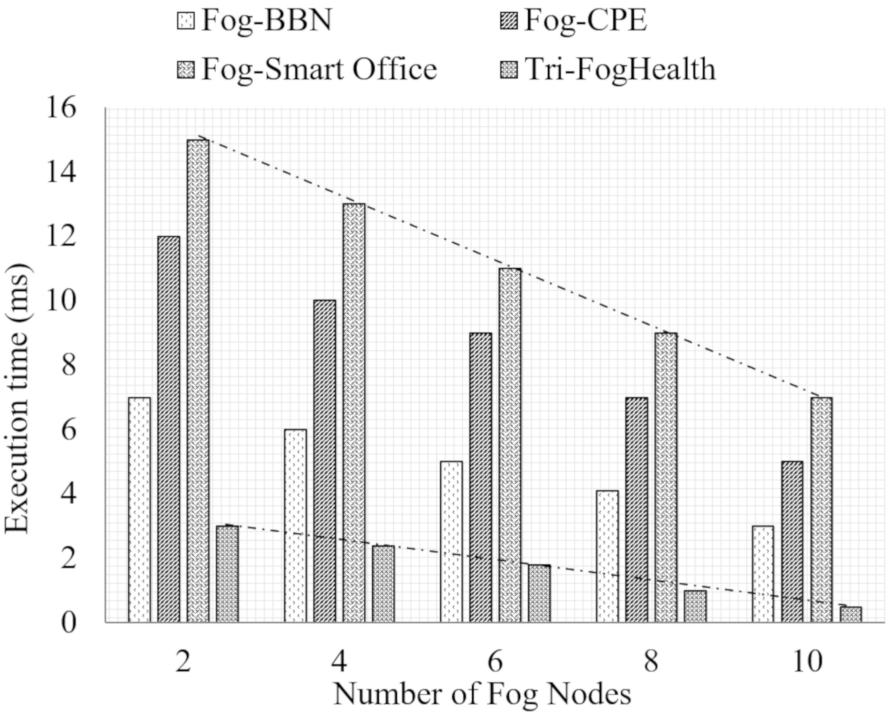


Figure 10. Comparison on Execution Time upon Number of Fog Nodes.

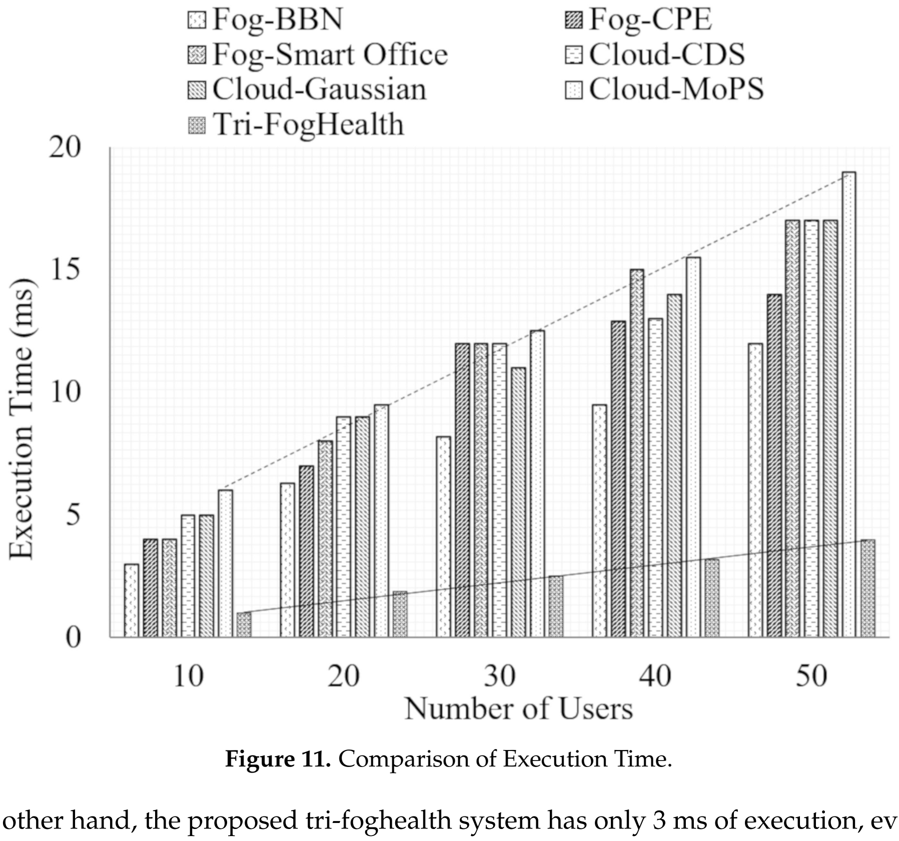


Figure 11. Comparison of Execution Time.

On the other hand, the proposed tri-foghealth system has only 3 ms of execution, even with two fog nodes. And the execution time is 0.5 ms in the presence of 10 fog nodes. This analysis shows that the fault data and redundant data elimination phases play a pivotal role in analyzing health data. With both approaches, we achieve timely detection on the user's health status.

5.3.3. Analysis of Detection Accuracy

Detection accuracy is defined as the number of corrective actions taken by the fog layer after analyzing the user's health status. Accurate detection is the ultimate goal of any smart healthcare system.

In Figure 12, the detection accuracy achieved by tri-foghealth work is compared with all prior research works concerning the number of users. In Figure 13, the detection accuracy is compared with fog based approach based on the number of fog nodes. In both analyzes, we can see that the tri-foghealth system has only slight variations. It maintains the accuracy curve more significantly than 95%.

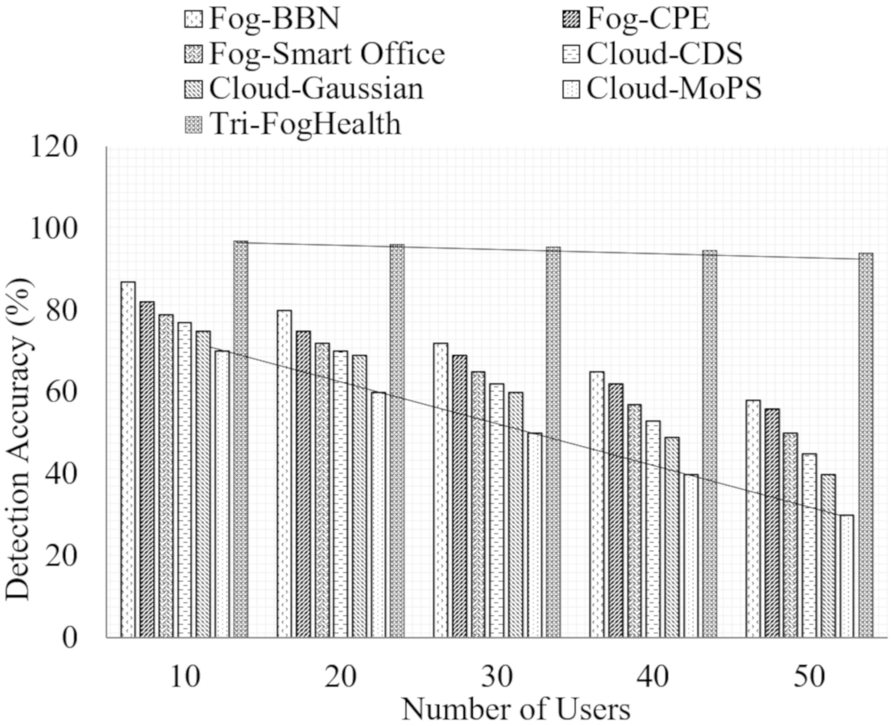


Figure 12. Comparison of Detection Accuracy.

The other methods have lower detection accuracy when the number of users increases. The existing accuracy curve shows that accuracy is decreased linearly. The main reason behind this much accuracy degradation is the algorithms used by the existing works are inefficient. In most cases, the Bayesian classifier is proposed, which has limitations in constructing the classifier model. Besides, most of the current works use the biosensor data for the user's health status detection. For accurate detection, behavior and environmental data are required. Thus, the lack of these data in existing works traps the detection accuracy to 30%. Even for 50 users, the tri-foghealth system achieves 94% in detection accuracy.

On the other hand, accuracy concerning the number of fog nodes shows that the proposed work achieves better accuracy even with two fog nodes. As we suppressed fault data and redundant data in the wearable layer, we make better accuracy greater than 90%, while other fog-based methods acquire lower accuracy, lower than 60%. In addition, the data overloading of fog nodes is a major problem in fog-based ways. Analyzing a massive amount of data in a single fog node makes the fog node inefficient. The presence of the health off procedure in proposed work supports at a high accuracy level.

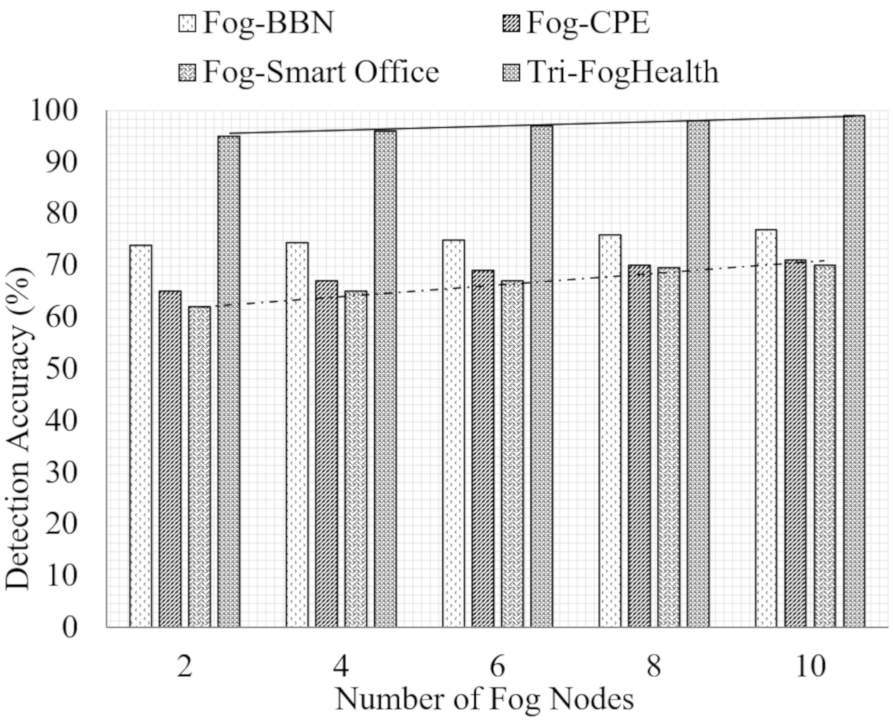


Figure 13. Comparison of Detection Accuracy.

5.3.4. System Stability

System stability is defined as the ability of the healthcare system to support a large number of users without a reduction in its efficiency and accuracy. In precise terms, it validates the scalability of the proposed healthcare system.

The comparative analysis of system stability is shown in Figure 14. Comparison is made for fog-based systems since the cloud-based systems can support a more significant number of users due to resource availability. The main drawback of cloud-based systems is high latency. Although fog-based systems resolve these problems, the stability of the system is still a concern. Since the fog nodes generally have resources lower than the cloud server and most of the users in the same region access the same fog nodes. In this analysis, we consider 50 users in the system.

In the presence of two fog nodes, most of the works have nearly 75% of stability. The stability is then increased linearly concerning the number of fog nodes. However, the highest stability achieved by the existing works is lower than 80% because all these works have an imbalanced load among fog nodes. As a result, the overloading fog nodes become unable to process data requests. On the other hand, the proposed tri-foghealth system achieves 96% for users, even with two fog nodes. The main reason behind these results is (i) fault data elimination by PDA, (ii) redundant data elimination by data on-looker, and (iii) health off by using the MoSHO algorithm. Altogether, tri-foghealth achieves system stability up to 98%. Form the comparison; we can conclude that the proposed tri-foghealth system has much better efficiency.

In Figure 15, system stability is compared between fog based and cloud-based approaches concerning the number of users. For a smart healthcare system, it is necessary to process many users without compromising accuracy. The analysis shows that the proposed foghealth approach achieves better stability regardless of the number of users (i.e.,) it reaches 96% and above strength for a varying number of users. An increase in the number of users and amount of data does not affect the stability of the proposed system. In this case, different fog based and cloud-based approaches achieve the same level of stability, which lies between 65% to 80%.

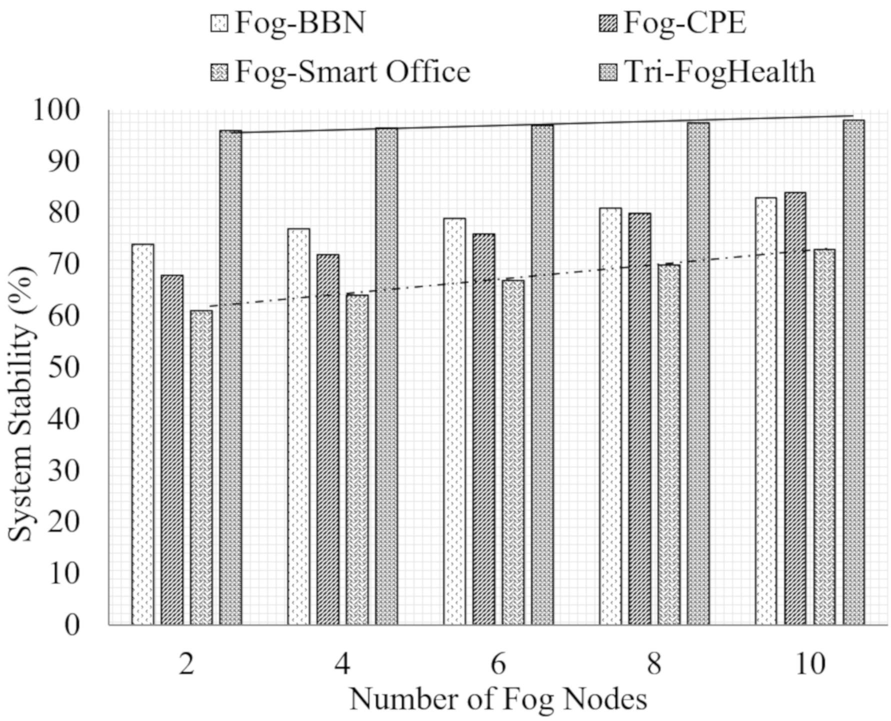


Figure 14. Comparison of System Stability.

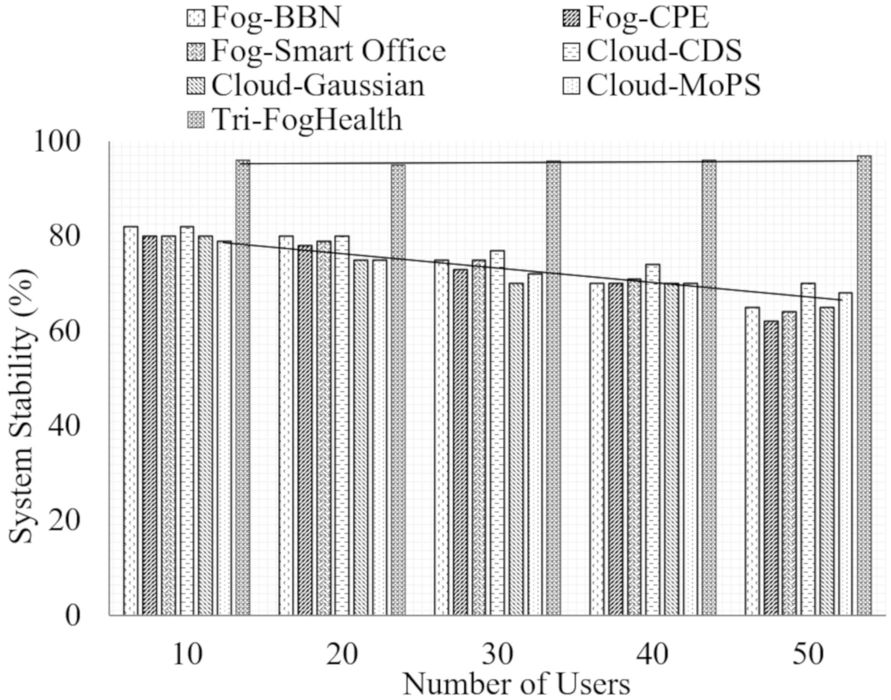


Figure 15. Comparison of System Stability based on Number of Users.

Further, there is a gradual decrease in stability with an increase in the number of users. An increase in the number of users increases the amount of data to be processed in fog and cloud layers. Although the cloud is more scalable than fog nodes, algorithm design also matters in achieving system stability. This analysis shows that the proposed tri-foghealth approach and the algorithms involved are effectual in analyzing the user's health status.

5.3.5. Discussion on Results

In this subsection, we discuss our proposed system. First, we summarize the average results of the proposed tri-foghealth system in Table 7. In this table, we have compared the proposed and existing works based on the average values obtained for each metric. The comparison shows that the proposed work achieves much better performance compared to existing research works. Mainly, the latency and execution time is relatively low in the proposed system without loss in detection accuracy. It shows that the proposed tri-foghealth system is suitable for real-time healthcare analysis.

Table 7. Numerical Comparison on Obtained Results.

Parameter	**Cloud-CDS [36]**	**Cloud-Gaussian [37]**	**Cloud-MoPS [38]**	**Fog-BBN [39]**		**Fog-CPE [40]**		**Fog-Smart Office [41]**		**Tri-FogHealth**	
				WRT n	**WRT m**	**WRT n**	**WRT m**	**WRT n**	**WRT m**	**WRT n**	**WRT m**
Latency (ms)	18.52	17.4	16.4	7.8	6.96	9.98	7.92	11.2	9.02	3.44	2.9
Execution Time (ms)	11.2	11.2	12.5	7.8	5.02	9.98	8.6	11.2	11	2.52	1.74
Detection Accuracy (%)	61.6	58.6	50	72.4	75.3	68.8	68.4	64.6	66.7	95.44	97
System Stability (%)	76.6	72	72.8	78.8	74.4	76	72.6	67	73	97	96

In this subsection, we discuss our proposed system. First, we summarize the average results of the proposed tri-foghealth system in Table 7. The comparison shows that the proposed work achieves much better performance compared to existing research works. Mainly, the latency and execution time is relatively low in the proposed system without loss in detection accuracy. It shows that the proposed tri-foghealth system is suitable for real-time healthcare analysis.

In Table 8, we summarize the overall contribution made in this work and its impact on results. We can see that the proposed algorithm supports the system to achieve better results in terms of latency, execution time, detection accuracy, and system stability. Although the proposed approach performs better efficiency, it also achieves tolerable complexity.

Table 8. Summarization of Proposed System.

The Methodology Used in Tri-FogHealth	**Contribution in Results**
RK-PCA Fault Detection	Eliminates fault data generated by wearable devices Increases detection accuracy since it allows only proper data for analysis Minimizes execution time since a large amount of fault data is suppressed
FaMOORA Redundant Data Elimination	Eliminates redundant data generated by wearable devices Minimizes execution time since the repeated data has been removed
2L-2HMM Health Status Prediction	Predicts user's health status based on temporal variations Increases detection accuracy since it predicts the user's next state Minimizes latency
SpikQ-Net Health Status Detection	Detect user's current health status Minimizes latency since the response is generated in the fog layer Detection accuracy is high since it considers there categories of attributes Execution time is low as the SpikQ-Net which is a machine learning technique that works faster than statistical approaches
MoSHO Health Off	Enable offloading procedure when the fog becomes overloaded Minimizes latency by offloading data to optimal fog nodes Increases system stability by maintaining a balanced load among the fog

The complexity analysis of the proposed system is summarized as follows,

$$Complexity = nT + (n - FD)T + 2T + T$$

As in the above, the complexity is also low in the proposed system. Here, n represents the number of data instances, and FD defines fault data. When compared to Gaussian and BBN classifiers, the complexity is low in the tri-foghealth system.

6. Conclusions

In this paper, we propose a novel tri-foghealth system for the smart healthcare system. The tri-foghealth system has worked upon three main layers by using wearable technology. The biomedical data generated from the wearable layer is processed in PDA to eliminate fault data using the RK-PCA algorithm. Then, the faultless data is fed into the data on-looker node in the intelligent fog layer; the data on-looker node is responsible for predicting the user's health status. For this, it uses temporal data in the 2L-2HMM algorithm. Before prediction, the current data instance is subjected to redundant data elimination process by the FaMOORA algorithm. After health status prediction, the data is processed by the fog nodes using SpikQ-Net, which works upon behavior, biomedical, and environmental attributes. Based on the emergency level of data, the immediate action is taken in the fog layer. Simultaneously, the health off procedure followed with the MOSHO algorithm to prevent the intelligent fog layer from overloading. The experimental results show that the proposed tri-foghealth system outperforms the current research works. In the future, we have planned to test the tri-foghealth system for specific disease diagnosis as the present system is designed to find general abnormalities. In addition, we intend to extend this healthcare system with trusted services as the health data is relatively sensitive. Thus, security provisioning is also a better research direction to extend the tri-foghealth system.

Author Contributions: Conceptualization, M.I. and L.L.; methodology, M.I.; software, M.I.; validation, M.I.; L.L. and A.N.; formal analysis, M.I.; L.L. and A.K.; investigation, L.L.; resources, G.L. and L.L.; data curation, G.L. and H.W.; writing—original draft preparation, M.I.; writing—review and editing, M.I.; A.N. and A.K.; visualization, M.I and Y.M.A.; supervision, L.L.; project administration, L.L.; funding acquisition, A.M.E.-S. All authors have read and agreed to the published version of the manuscript.

Funding: This research supported by King Saud University with grand of Researchers Supporting Project number (RSP-2020/133).

Acknowledgments: The authors extend their appreciation to King Saud University for funding this work through Researchers Supporting Project number (RSP-2020/133), King Saud University, Riyadh, Saudi Arabia.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Mohapatra, S.; Mohanty, S.; Mohanty, S. Smart healthcare: An approach for ubiquitous healthcare management using IoT. In *Advances in Ubiquitous Sensing Applications for Healthcare*; Academic Press: Cambridge, MA, USA, 2019.
2. Ahad, A.; Tahir, M.A.; Yau, K.A. 5G-based smart healthcare network: Architecture, taxonomy, challenges, and future research directions. *IEEE Access* **2019**, *7*, 100747–100762. [CrossRef]
3. Tian, S.; Yang, W.; Grange, J.M.; Wang, P.; Huang, W.; Ye, Z. Smart healthcare: Making medical care more intelligent. *Glob. Health J.* **2019**, *3*, 62–65. [CrossRef]
4. Muzny, M.; Henriksen, A.; Giordanengo, A.; Mužík, J.; Grøttland, A.; Blixgård, H.; Hartvigsen, G.; Årsand, E. Wearable sensors with possibilities for data exchange: Analyzing status and needs of different actors in mobile health monitoring systems. *Int. J. Med. Inform.* **2019**, *133*, 104017. [CrossRef] [PubMed]
5. Papa, A.; Mital, M.; Pisano, P.; Giudice, M.D. E-health and wellbeing monitoring using smart healthcare devices: An empirical investigation. *Technol. Forecast. Soc. Chang.* **2020**, *153*, 119226. [CrossRef]
6. Mutlag, A.A.; Ghani, M.K.; Arunkumar, N.; Mohammed, M.A.; Mohd, O. Enabling technologies for fog computing in healthcare IoT systems. *Future Gener. Comput. Syst.* **2019**, *90*, 62–78. [CrossRef]
7. Mahmud, R.; Ramamohanarao, K.; Buyya, R. Application management in fog computing environments: A taxonomy, review, and future directions. *ACM Comput. Surv.* **2020**. [CrossRef]
8. Zhao, R.; Yan, R.; Chen, Z.; Mao, K.; Wang, P.; Gao, R.X. Deep Learning and Its Applications to Machine Health Monitoring: A Survey. *arXiv* **2016**, arXiv:abs/1612.07640. [CrossRef]

9. Soualhi, M.; Nguyen, K.T.; Soualhi, A.; Medjaher, K.; Hemsas, K.E. Health monitoring of bearing and gear faults by using a new health indicator extracted from current signals. *Measurement* **2019**, *141*, 37–51. [CrossRef]
10. Fu, Y.; Peng, C.; Gomez, F.; Narazaki, Y.; Spencer, B.F. Sensor fault management techniques for wireless smart sensor networks in structural health monitoring. *Struct. Control Health Monit.* **2019**. [CrossRef]
11. Petrovska, B.; Zdravevski, E.; Lameski, P.; Corizzo, R.; Štajduhar, I.; Lerga, J. Deep learning for feature extraction in remote sensing: A case-study of aerial scene classification. *Sensors* **2020**, *20*, 3906. [CrossRef]
12. Munir, M.; Siddiqui, S.A.; Chattha, M.A.; Dengel, A.; Ahmed, S. FuseAD: Unsupervised anomaly detection in streaming sensors data by fusing statistical and deep learning models. *Sensors* **2019**, *19*, 2451. [CrossRef]
13. Merrill, N.; Eskandarian, A. Modified autoencoder training and scoring for robust unsupervised anomaly detection in deep learning. *IEEE Access* **2020**, *8*, 101824–101833. [CrossRef]
14. Ceci, M.; Corizzo, R.; Japkowicz, N.; Mignone, P.; Pio, G. ECHAD: Embedding-based change detection from multivariate time series in smart grids. *IEEE Access* **2020**, *8*, 156053–156066. [CrossRef]
15. Hu, J.; Wu, K.; Liang, W. An IPv6-based framework for fog-assisted healthcare monitoring. *Adv. Mech. Eng.* **2019**, *11*. [CrossRef]
16. Jeyaraj, P.R.; Rajan, S.E.; Martis, R.J.; Panigrahi, B.K. Fog computing employed computer aided cancer classification system using deep neural network in internet of things based healthcare system. *J. Med. Syst.* **2019**, *44*, 34.
17. Nweke, H.F.; Wah, T.Y.; Mujtaba, G.; Al-garadi, M.A. Data fusion and multiple classifier systems for human activity detection and health monitoring: Review and open research directions. *Inf. Fusion* **2019**, *46*, 147–170. [CrossRef]
18. Goschenhofer, J.; Pfister, F.M.; Yuksel, K.A.; Bischl, B.; Fietzek, U.; Thomas, J. Wearable-based Parkinson's Disease Severity Monitoring using Deep Learning. *arXiv* **2019**, arXiv:1904.10829.
19. Mani, N.; Singh, A.; Nimmagadda, S.L. An IoT guided healthcare monitoring system for managing real-time notifications by fog computing services. *Procedia Comput. Sci.* **2020**, *167*, 850–859. [CrossRef]
20. Zheng, H.; Xiong, K.; Fan, P.; Zhong, Z.; Letaief, K.B. Fog-assisted multiuser swipt networks: Local computing or offloading. *IEEE Internet Things J.* **2019**, *6*, 5246–5264. [CrossRef]
21. Runkle, J.; Sugg, M.; Boase, D.; Galvin, S.L.; Coulson, C.C. Use of wearable sensors for pregnancy health and environmental monitoring: Descriptive findings from the perspective of patients and providers. *Digit. Health* **2019**, *5*. [CrossRef]
22. Sangeetha, A.; Thangavel, A. Pervasive healthcare system based on environmental monitoring. *Intell. Pervasive Comput. Syst. Smarter Healthc.* **2019**. [CrossRef]
23. Aboudi, N.E.; Benhlima, L. Big data management for healthcare systems: Architecture, requirements, and implementation. *Adv. Bioinform.* **2018**. [CrossRef]
24. Sun, F.; Zang, W.; Gravina, R.; Fortino, G.; Li, Y. Gait-based identification for elderly users in wearable healthcare systems. *Inf. Fusion* **2020**, *53*, 134–144. [CrossRef]
25. Albahri, A.S.; Albahri, O.S.; Zaidan, A.A.; Zaidan, B.B.; Hashim, M.; Alsalem, M.A.; Mohsin, A.H.; Mohammed, K.I.; Alamoodi, A.H.; Enaizan, O.; et al. Based on multiple heterogeneous wearable sensors: A smart real-time health monitoring structured for hospitals distributor. *IEEE Access* **2019**, *7*, 37269–37323. [CrossRef]
26. Zhang, H.; Liu, J.; Kato, N. Threshold tuning-based wearable sensor fault detection for reliable medical monitoring using bayesian network model. *IEEE Syst. J.* **2018**, *12*, 1886–1896. [CrossRef]
27. Colopy, G.W.; Roberts, S.J.; Clifton, D.A. Bayesian optimization of personalized models for patient vital-sign monitoring. *IEEE J. Biomed. Health Inform.* **2018**, *22*, 301–310. [CrossRef]
28. Liu, Y.; Zhang, L.; Yang, Y.; Zhou, L.; Ren, L.; Wang, F.; Liu, R.; Pang, Z.; Deen, M.J. A novel cloud-based framework for the elderly healthcare services using digital twin. *IEEE Access* **2019**, *7*, 49088–49101. [CrossRef]
29. Anagaw, A.; Chang, Y. A new complement naïve Bayesian approach for biomedical data classification. *J. Ambient Intell. Humaniz. Comput.* **2019**, *10*, 3889–3897. [CrossRef]
30. Pathinarupothi, R.K.; Durga, P.; Rangan, E. IoT-based smart edge for global health: Remote monitoring with severity detection and alerts transmission. *IEEE Internet Things J.* **2018**, *6*, 2449–2462. [CrossRef]
31. Rahmani, A.M.; Gia, T.N.; Negash, B.; Anzanpour, A.; Azimi, I.; Jiang, M.; Liljeberg, P. Exploiting smart e-Health gateways at the edge of healthcare Internet-of-Things: A fog computing approach. *Future Gener. Comput. Syst.* **2018**, *78*, 641–658. [CrossRef]

32. Gai, K.; Lu, Z.; Qiu, M.; Zhu, L. Toward smart treatment management for personalized healthcare. *IEEE Netw.* **2019**, *33*, 30–36. [CrossRef]
33. Hassan, M.K.; El-Desouky, A.I.; Badawy, M.M.; Sarhan, A.M.; Elhoseny, M.; Manogaran, G. EoT-driven hybrid ambient assisted living framework with naïve Bayes–firefly algorithm. *Neural Comput. Appl.* **2018**, *31*, 1275–1300. [CrossRef]
34. Tuli, S.; Basumatary, N.; Gill, S.S.; Kahani, M.; Arya, R.C.; Wander, G.S.; Buyya, R. HealthFog: An ensemble deep learning based smart healthcare system for automatic diagnosis of heart diseases in integrated IoT and fog computing environments. *Future Gener. Comput. Syst.* **2020**, *104*, 187–200. [CrossRef]
35. Muhammed, T.; Mehmood, R.; Albeshri, A.A.; Katib, I.A. UbeHealth: A personalized ubiquitous cloud and edge-enabled networked healthcare system for smart cities. *IEEE Access* **2018**, *6*, 32258–32285. [CrossRef]
36. Khattak, H.A.; Arshad, H.; Islam, S.U.; Ahmed, G.; Jabbar, S.; Sharif, A.M.; Khalid, S. Utilization and load balancing in fog servers for health applications. *Eurasip J. Wirel. Commun. Netw.* **2019**, *2019*, 1–12. [CrossRef]
37. Mutlag, A.A.; Ghani, M.K.; Mohammed, M.A.; Maashi, M.S.; Mohd, O.; Mostafa, S.A.; Abdulkareem, K.H.; Marques, G.; Díez, I.D. MAFC: Multi-Agent fog computing model for healthcare critical tasks management. *Sensors* **2020**, *20*, 1853. [CrossRef]
38. Sharma, S.; Saini, H. A novel four-tier architecture for delay aware scheduling and load balancing in fog environment. *Sustain. Comput. Inform. Syst.* **2019**, *24*, 100355. [CrossRef]
39. Adhikari, M.; Gianey, H.K. Energy efficient offloading strategy in fog-cloud environment for IoT applications. *Internet Things* **2019**, *6*, 100053. [CrossRef]
40. Naghshvarianjahromi, M.; Kumar, S.; Deen, M.J. Brain-inspired intelligence for real-time health situation understanding in smart e-health home applications. *IEEE Access* **2019**, *7*, 180106–180126. [CrossRef]
41. Zhu, T.; Colopy, G.W.; MacEwen, C.; Niehaus, K.E.; Yang, Y.; Pugh, C.W.; Clifton, D.A. Patient-specific physiological monitoring and prediction using structured gaussian processes. *IEEE Access* **2019**, *7*, 58094–58103. [CrossRef]
42. Leu, F.; Ko, C.; You, I.; Choo, K.R.; Ho, C. A smartphone-based wearable sensors for monitoring real-time physiological data. *Comput. Electr. Eng.* **2018**, *65*, 376–392.
43. Verma, P.; Sood, S.K. Fog assisted-IoT enabled patient health monitoring in smart homes. *IEEE Internet Things J.* **2018**, *5*, 1789–1796. [CrossRef]
44. Dautov, R.; Distefano, S.; Buyya, R. Hierarchical data fusion for smart healthcare. *J. Big Data* **2019**, *6*, 1–23. [CrossRef]
45. Bhatia, M.; Sood, S.K. Exploring temporal analytics in fog-cloud architecture for smart office healthcare. *Mob. Netw. Appl.* **2019**, *24*, 1392–1410. [CrossRef]
46. Brauers, W.K.; Zavadskas, E.K. The MOORA method and its application to privatization in a transition economy. *Control Cybern.* **2006**, *35*, 445–469.
47. Xiang, S.; Zhang, Y.; Gong, J.; Guo, X.; Lin, L.; Hao, Y. STDP-based unsupervised spike pattern learning in a photonic spiking neural network With VCSELs and VCSOAs. *IEEE J. Sel. Top. Quantum Electron.* **2019**, *25*, 1–9. [CrossRef]
48. Salahshour, E.; Malekzadeh, M.; Gholipour, R.; Khorashadizadeh, S. Designing multi-layer quantum neural network controller for chaos control of rod-type plasma torch system using improved particle swarm optimization. *Evol. Syst.* **2019**, 1–15. [CrossRef]
49. Dhiman, G.; Kumar, V. Spotted hyena optimizer: A novel bio-inspired based metaheuristic technique for engineering applications. *Adv. Eng. Softw.* **2017**, *114*, 48–70. [CrossRef]

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.