sustainability

MDPI

Article

An Intelligent Health Care System in Fog Platform with Optimized Performance

Subhranshu Sekhar Tripathy [1,2,*], Mamata Rath [2], Niva Tripathy [2], Diptendu Sinha Roy [1], John Sharmila Anand Francis [3] and Sujit Bebortta [2]

[1] Department of Computer Science and Engineering, National Institute of Technology, Meghlaya 793003, India
[2] Department of Computer Science and Engineering, DRIEMS Autonomous College, Cuttack 754025, India
[3] Department of Computer Science, King Khalid University, Abha 62529, KSA, Saudi Arabia
* Correspondence: subhranshutripathy@driems.ac.in or p19cs013@nitm.ac.in

Abstract: Cloud computing delivers services through the Internet and enables the deployment of a diversity of apps to provide services to many businesses. At present, the low scalability of these cloud frameworks is their primary obstacle. As a result, they are unable to satisfy the demands of centralized computer systems, which are based on the Internet of Things (IoT). Applications such as disease surveillance and tracking and monitoring systems, which are highly latency sensitive, demand the computation of the Big Data communicated to centralized databases and from databases to cloud data centers, resulting in system performance loss. Recent concepts, such as fog and edge computing, offer novel approaches to data processing by relocating the processing power and other resources closer to the end user, thereby reducing latency and maximizing energy efficiency. Existing fog models, on the other hand, have a number of limitations and tend to prioritize either the precision of their findings or a faster response time, but not both. For the purpose of applying a healthcare solution in the real world, we developed and implemented a one-of-a-kind architecture that integrates quartet deep learning with edge computing devices. The paradigm that has been developed delivers health management as a fog service through the Internet of Things (IoT) devices and efficiently organizes the data from patients based on the requirements of the user. FogBus, a fog-enabled cloud framework, is used to measure the effectiveness of the proposed structure in regards to resource usage, network throughput, congestion, precision, and runtime. To maximize the QoS or forecast the accuracy in different fog computing settings and for different user requirements, the suggested technique can be set up to run in a number of different modes.

Keywords: IoT; fog computing; cloud computing; deep learning; heart disease

Citation: Tripathy, S.S.; Rath, M.; Tripathy, N.; Roy, D.S.; Francis, J.S.A.; Bebortta, S. An intelligent Health Care System in Fog Platform with Optimized Performance. *Sustainability* **2023**, *15*, 1862. https://doi.org/10.3390/su15031862

Academic Editors: Amir Masoud Rahmani, Stavros Shiaeles, Firuz Kamalov, Seyedeh Yasaman Hosseini Mirmahaleh and Hao-Chiang Koong Lin

Received: 16 August 2022
Revised: 27 December 2022
Accepted: 13 January 2023
Published: 18 January 2023

1. Introduction

The IoT has seen widespread adoption in the healthcare industry in recent years, particularly in applications using ubiquitous sensors and actuators interacting via Wireless Sensor Networks (WSN) and systems for real-time data analysis and suggestion [1]. The cloud computing approach uses the internet to deliver on-demand services to cloud users and has become an essential part of the modern economy [2]. Currently, the cloud has become the ultimate use of computing and is gaining a lot of attention from businesses and schools. At present, thanks to the provision of services and infrastructure, cloud computing can enable new application models such as IoT, fog computing, edge computing, as well as Big Data [3,4]. New study fields are being created for cloud computing due to technological breakthroughs such as edge computing, fog computing, mist computing, the IoT, and smart cities [3]. To enable fog computing to meet the demand of latency-sensitive or real-time applications, these developing technologies offer storage space and computing with transmission edge devices. This facilitates enhanced portability and security, and reduced latency and network throughput [5,6]. Fog computing reduces the network latency and reaction time by placing computing resources, such as switches and routers, gateways, and mobile base stations, in densely populated areas.

Latency and response time have been found to be the utmost critical factors, as well as the greatest challenges, to optimizing Quality of Service (QoS) specifications in real-time fog scenario, according to research carried out by Mutlag et al. [7]. They investigated these difficulties of fog computing in the context of healthcare applications. Fog computing has been implemented in healthcare, which has resulted in encouraging developments in this area. The healthcare industry is one of the most prominent application sectors that require precise and real-time data. Through the use of fog computing, resources can move nearer their respective users, reducing latency and, as a result, increasing the level of security. However, the faster delivery of findings is insufficient because, with the data being so sensitive, we cannot make any concessions regarding the correctness of the results. Utilizing state-of-the-art analysis software, such as those that make use of deep learning and its variants, which have been trained on a vast dataset, is one method for achieving high levels of accuracy in one's work.

Deep learning [8] has grown in popularity over the past few decades, particularly in the fields of object recognition [6] and voice recognition [8]. Ensemble learning is used to combine the best findings of various classifiers [9]. The bagging classifier ensemble trains the base classifier on the random subsets of data, then aggregates their predictions by voting or by average. Randomization in dataset distribution reduces variance compared to a single estimator. In order to train and forecast accurately, modern healthcare deep learning models are computationally intensive and resource intensive [10]. Sophisticated neural networks take time to train and interpret data. Training and using these complicated neural networks to interpret data takes a substantial amount of time. Higher accuracy requires a more advanced network and longer prediction time [11]. Healthcare and other IoT applications that necessitate real-time findings have needed help. As computation on the edge reduces turnaround time, merging complicated ensemble deep learning models with edge computing to achieve high-accuracy real-time outputs is a new research field. This work intends to address this gap and create a computing framework which leverages edge resources for low latency and deep learning frameworks for high accuracy. To minimize the result delivery time, processing has been brought to Edge devices nearer to the patient. Some of these works rely on simulations [12] and need a deployable framework.

Conventional healthcare systems in which fog or cloud computing platforms are integrated with IoT collect patient's information in a timely manner through pre-configured devices. Predictions of cardiac issues have been the focus of numerous prior works' attempts to utilize IoT. However, due to the strict constraints enforced by medical standards agencies, they have not yet been successful. These laws require a particular level of accuracy. As deep learning has grown in popularity in recent years, more current technologies have even outperformed medical professionals in the accuracy of heart disease diagnosis [13,14]. This research intends to merge deep learning and IoT in the healthcare sector to persuade medical standards agencies to adopt a reduced latency, high-precision model to assist in easing the current doctor shortage. Only some works, such as [15], seek to combine these two paradigms. None of these initiatives, however, take advantage of edge computing's decentralized nature to boost performance through the deployment of aggregate deep-learning models.

An engrained IoT, fog computing, and cloud computing based computation framework is needed to provide efficient computing services to heart patients, as well as for other applications that demand real-time results. These services must be provided to users who require real-time results. This work was motivated by the need for more models or frameworks that combine high-accuracy deep learning models with edge computing nodes' low latency.

Using deep learning and the IoT, our cloud-based smart healthcare system, HealthFog, can automatically diagnose cardiac conditions. HealthFog is a healthcare platform that provides a less invasive fog service and efficiently stores vital patient data. These numbers are collected from a wide range of IoT gadgets.

The important aspects of this paper are:

- It has been suggested that fog computing could be used as an example of a generalized system design for the advancement of collaborative deep learning.
- HealthFog uses deep learning to automatically analyze cardiac patient data.
- HealthFog was deployed with the FogBus framework so that it could integrate with IoT-edge-cloud to perform real-time data processing.
- Performance characteristics, such as accuracy, reaction time, and network capacity, are demonstrated and studied for the HealthFog deployment.

The remaining sections of the paper are laid out as follows: the existing healthcare systems and related projects are presented in Section 2; Section 3 presents the proposed model; Section 4 illustrates the components of the model; Section 5 details the implementation of the proposed work; Section 6 details the comparative analysis; and Section 7 illustrates the conclusion and future scope.

2. Literature Review

Fog computing is an innovative technology for analyzing IoT-generated healthcare data. Fog computing can manage cardiac patient data at edge devices or fog nodes with large processing capacities to reduce the delay. Edge devices are closer to IoT than cloud data centres.

Considerable work has gone into creating IoT–based solutions for remote health monitoring. Continuous health monitoring using tailored 6LoWPAN is proposed by Gia et al. [16]. The technology provides effective real-time, remote ECG monitoring via an established network.

Patients with chronic conditions can benefit from Gomez et al.'s [17] introduction of an IoT-based patient monitoring system, which tracks their current health and advise them on improving it through exercise. Not only does the system gather bio-signals (such as an electrocardiogram), but it also collects environmental information (i.e., time and location). With the use of an Android app, both the doctor and the patient will have access to the data that has been gathered.

A comprehensive information and communication technology system has been presented by Fanucci et al. [18] to monitor patients at home. The ECG, SpO2, weight, and blood pressure of a patient are all collected by the system using biomedical sensors. The captured data are then sent to the hospital's information system so that remote monitoring can occur. As it can assist in the early detection of changes in patients' vital signs, the system contributes to a decrease in the cases of subsequent hospitalizations.

The authors offer an intelligent healthcare system using the IoT [19]. This setup can track various signals, including glucose level, ECG, blood pressure, body temperature, and SpO2, and then wirelessly transfer the gathered information to Raspberry Pie using Zigbee. Through a smartphone application, end-users, such as doctors and caregivers, can watch the data being collected.

Mahmud et al. [20] propose a Fog-based IoT-Healthcare (FIH) integrated design and analyze the CloudFog services in the compatible healthcare systems. IFogSim [21] is used to assess FIH's energy usage and response time. Speed and precision measure FIH's efficiency.

Using patient EHRs in the Fast Healthcare Interoperability Resources (FHIR) format, Alvin et al. [22] suggested a Scalable and Accurate deep learning (SADL) model. Without the need for site-specific data harmonization, the deep learning methods in the SADL model can accurately predict a wide variety of medical events from various locations. In addition, the proposed method is validated with de-identified Electronic Health Record

(EHR) data from two US academic medical facilities with 216,221 adult patients hospitalized for at least 24 h, thus enhancing the prediction accuracy.

To offer humans contextual information and monitor their vital signs using a robot assistant, Pham et al. [23] developed a Cloud-based Smart Home Environment (CoSHE) to deliver home healthcare. CoSHE initially employs non-invasive, wearable sensors to obtain audio, motion, and physiological signs in order to subsequently provide this context of the residents' everyday activities. The effectiveness of CoSHE is evaluated via a case study of robotic assistance using Google APIs. CoSHE, on the other hand, is a generic healthcare application that collects and processes patient data on a modest scale without data analytics. Its effectiveness in the QoS parameters has not been validated on a proper cloud system.

Alam et al. [24] suggested Edge-of-Things Computation (EoTC) to reduce healthcare data processing costs. In addition, a distributed provisioning technique based on the Alternating Direction Method of Multipliers (ADMM) is described, along with a portfolio optimization strategy for choosing virtual machines (VMs) to handle healthcare data. Further, the experimental data demonstrate that the EoTC paradigm performs better than the proposed methodology with regard to cost, despite the fact that the EoTC paradigm does not permit a performance evaluation based on the QoS considerations.

Sanaz et al. [25] presented an end-to-end security strategy for a mobility-enabled healthcare IoT that uses DTLS to protect the transmission across intelligent gateways without device layer reconfiguration. The suggested technique reduces communication costs by 26% and delays by 16% when applied in Cooja, a simulation environment.

Information technology will allow healthcare providers and patients to improve their experiences and services due to the increased availability and real-time data interchange. Despite these promising technological advances, data integrity and consistency still need to be solved. Thus, engineering a fix is crucial. This paper [26] proposes two models for using fog computing in healthcare: private fog computing distribution and public fog computing distribution. Each model has a unique scheme to assess malicious attack damage, correctly identify the affected transactions, and restore the destroyed data. Both models monitor system transactions using a transaction-dependency graph. These models worked after the examination.

The Smart e-Health Gateway exploits the gateways' strategic location at the network's edge to provide incorporated data mining, truly local computation, and file access. Creating an ageo-distributed intelligence layer between the cloud and sensor nodes provides scalability, power efficiency, and adaptability [3].

To leverage IoT-based fog computing in healthcare, it is necessary to find solutions to the following problems:

1. A real time application based on the concept of IoT in healthcare that can manage a huge amount of data collected from patients, with reduced latency and minimized power consumption.
2. Fog computing scenarios need a well-organized resource scheduling mechanism to accomplish user workloads and meet deadlines.
3. A deep learning-based fog computing algorithm can accurately assess heart disease severity.

3. Proposed Model

3.1. System Architecture

In order to highlight how the components can be dispersed across three levels and employed in smart hospitals, Figure 1 shows a thorough architecture of a IoT based health care system. Among these categories, the implanted sensors store this patient's health-related information. Assisting the patient allows for the private monitoring of many variables. Date, time, location, temperature, and other factors can be incorporated into this health information to enhance it. Circumstance awareness makes it possible to recognize a typical design and generates more precise inferences about the circumstances. For the information transfer from machines, equipment such as CAT scans and magnetic resonance imaging can also be used with various medical devices. The architecture of the machine contains the subsequent essential elements:

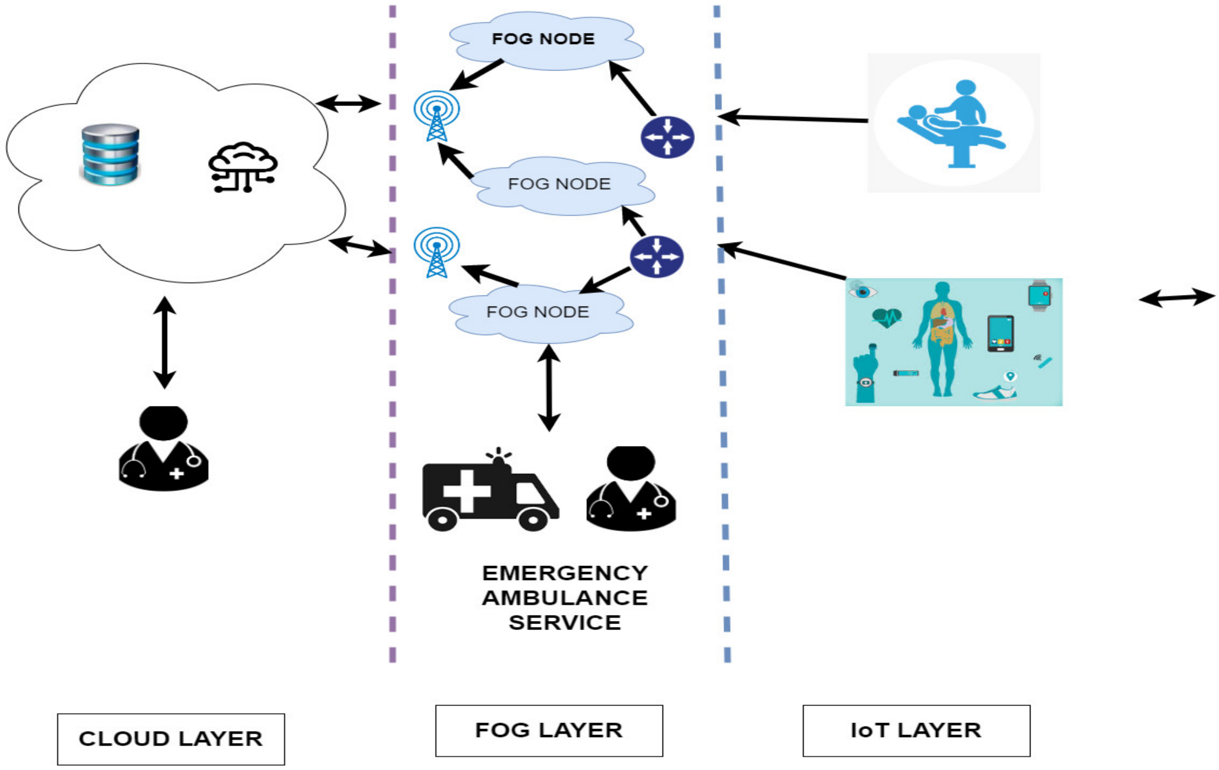

Figure 1. Proposed system architecture.

- Sensors and Actuators: Signals from this body and environment can be collected using the continuous identification, sensing, and transmission capabilities, as well as biomedical and perspective signals. Networks of intelligent e-Health, which use wireless or wired communication protocols, such as Bluetooth and Wi-Fi, to help transmit the data to the access.
- Gateways: This layer is formed after a number of intelligent e-Health gateways that are geographically distributed, similar to creating a simulated fog. Each gateway facilitates different transmission protocols by acting as a source of communication between the sensor nodes and the regional router or Web. The data from various subnetworks and the bearings' protocol conversion must be provided in order for the higher-level features, such as data aggregation, filtering, and dimensionality depletion, to be received [27–30].
- Back-End System: This back-end strategy results in a cloud computing program that runs streaming, data warehouse, and data analytics. Lastly, it can be used as an addition to the web client's GUI for the output and feedback. This available health, as well as environmental, data can be used as a source of huge data [31–36] for the statistical and epidemiological forecasting of upcoming epidemic diseases.

This gateway's primary job is to support various wireless features and control device-to-device communication. According to this section, we develop its specific component, which turns out to be a fog enabler, by building the orchestra by connecting gateways and putting in place various features, such as acting as a repository to momentarily store user data gathered by the sensors and fusing it with the other data using aggregation, interpretation techniques, and data fusion. These methods are crucial for the local pre-processing of sensor data and can be referred to as an intelligent e-Health Gateway.

3.2. Fog-Based Data Management and Analytics

As smart cities implement innovative IoT programs and try to glean more insights from regularly generated data in massive amounts to enhance or supply new smart services and respond to emergencies, they will need to employ new strategies and data management techniques. Traditional databases and business intelligence architectures will always be necessary for smart cities; however, IoT technologies require specialized competencies to manage varied data, continually, from different sources and in different forms. Data management is developing into a comprehensive field in the IoT context, covering preprocessing, batch, stream, and storage methodologies and platforms. This comprises several disciplines that deal with data, including data governance, data provisioning, data quality management, and data integration (propagation, consolidation, and federation). Consequently, handling IoT data within organizations, such as smart cities, is a difficult task, particularly when predictive analytics and decision-making involve the use of sizeable amounts of data from various heterogeneous sources. These companies use algorithms to filter data from diverse sources before it reaches a centralized information bank, delivering different levels of information quality. They also use automated data aggregation and classification techniques at the edge and in the fog to hasten the production of insights from data streams and safeguard data repositories against massive information volumes and high data velocity.

4. Fog Based Smart Healthcare Components

4.1. Hardware Component

- Network of Body Area Sensor devices: This element consists of three main sensing components: environmental sensors, activity sensors, and health sensors. Medical sensors include electrocardiogram (ECG) sensors, electroencephalogram (EEG) sensors, electromyogram (EMG) sensors, respiration sensors, thermocouple sensors, and differential pressure sensors with potentiometric sensors. Through related gateway devices, these components are in charge of the transmission of data from the patient's body.
- Gateway: Fog nodes, which include cell phones, laptops, and tablets, collect the data from the sensors throughout the environment and relay that information to the Broker/Worker units for in-depth analysis.
- Cloud Data Center: If the fog framework becomes saturated, services are latency-sensitive, or, if the length of the incoming data is significantly longer than usual, this fog-enabled innovative health makes use of the fog infrastructure or cloud data center resources.

4.2. Software Components

The relevant software components from the fog-enabled innovative health care model are data cleaning and processing, data analytics, resource supervising, the judgement module, the deep learning module, and the accumulating module. The relevant software components form the Fog enabled Smart health model are:

1. Data cleaning and pre-processing: preprocessing begins as soon as the data are submitted in order to filter the information, which also includes the use of data analytics technologies. In order to obtain the essential components of the data feature vectors that influence patient health conditions, the filtered data are condensed to a lower size, adopting Principal Component Analysis (PCA) and utilizing Set Partition [28,29] and are protected using the Singular Value Decomposition (SVD) method [30]. It immediately draws a conclusion from the input data, recommends medication and the relevant checkups based on the continuously trained healthcare experts who deal with the data, and then saves it in a database for future training, as needed.
2. Resource supervisor: the two elements that constitute this framework are the workload management and arbitration modules [27]. The task queues and job requests for

data processing are monitored by this workload manager. The fog or cloud resources that have been supplied for the processing tasks that have been planned and handled by this workload supervisor are assigned by the arbitration component. The Arbitration element, which is connected with this Broker end node, decides whether the cloud data center, the fog worker end node, or the fog computing node must provide this information in order to determine the results [27]. The main goal is to distribute the duties among several devices in order to balance the load and guarantee optimal performance. Fog-enabled innovative health permits users to tailor their load balancing and arbitration methods depending on the necessities of their applications.

3. Deep learning Module: The dataset is utilized to build a computational framework for locating data endpoints, which will contain the vectors created after pre-processing the data acquired from this wireless body area network. This forecasts and processes the output for the information acquired from these intelligent gateway devices based on the task assigned by the resource supervisor.
4. 4. Ensemble Module: This component estimates the output class, or if the individual has cardiac dysfunction, utilizes the polling data from a variety of models. The feature, which is hosted on the task's FogBus node, is responsible for conveying the accumulated dependable out-run through many worker end nodes.

5. Implementation

The demonstrated fog computing architecture in the preceding section collected the data of heart patients from the sensor and computed the results before transmitting them back to the data center, where the person has the coronary illness, with case probability. The directly visible gateway interface, the ensemble deep learning modules, and the data from the preparation modules are used in its execution. All of the elements listed in Section 3's various components were carried out using Python programming languages. To take advantage of the Python programming language, the implementation's preprocessing and deep learning with ensemble technique components were rearranged. Based on the spread, minimum, and maximum expected values of the area boundaries in the datasets, the preprocessing module uniformizes the data. Deep learning with the ensemble technique was implemented using the scikit learn Machine Learning Library in the Python environment [31]. The scikit learn module's bagging classifier approach was used to build the voting system in our suggested models. The deep learning network is employed as the classifier as the model base, and the number of classifiers are used as the input. As the model is being developed, the information is being dispersed randomly across the classifiers. It now accepts the predicted output of the core components and all of the anticipated classes as the input.

5.1. Data Processing Phase of Heart Patient

One must be prepared in advance to determine the estimations for numerous highlights of the contributions to deep learning models because the data from basic pule-oximeters or ECG devices is organized in a simple graphical manner [31,32]. The framework must take care of the domain information particular to each application's standardizing age information, as shown in Figure 1. In essence, patients with heart disease showed more severe hypertension than patients without heart disease, according to the data on resting blood pressure, which is comparably slow in both groups of patients. The communication of a healthy patient is leptokurtic, just as comprehending cholesterol levels reflects some objective, obvious behavior. In practice, even with the highest pulse, all patients who are deemed to be in healthy status have a much higher pulse rate, at approximately 160, compared to those who have heart disease, at approximately 150. Several criteria, particularly chest discomfort and fasting blood sugar, should be constantly altered from virtue to absolute virtue. Similarly to how the cardiac position recovered following the thallium test, the peak exercise ST section did as well.

5.2. Application of Deep Learning Ensemble

In this model, a deep Q network (DQN) ensemble method is applied for our careful application and prediction analysis. The model is used to address issues with binary classification. The Cleveland dataset's patient's cardiac continuous data were employed to develop the first model; this was subsequently utilized in order to approximate the outcomes from the persistent information input using the matching training model as shown in Figure 2.

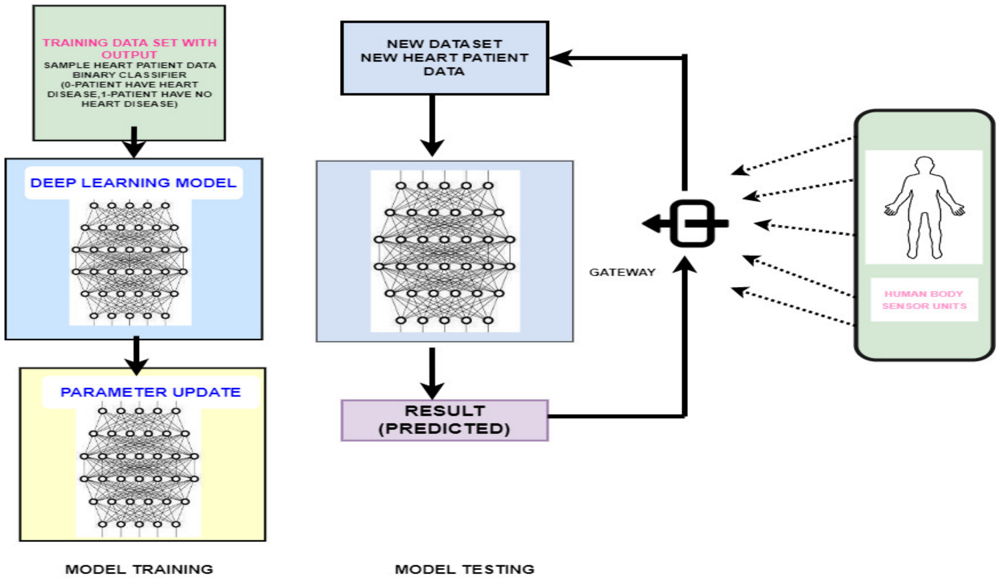

Figure 2. Deep learning module of proposed model.

This dataset is categorized into three parts for training, testing, and validation in the proportion 7:2:1. The dataset is classified into three categories: the training dataset, which composes 70% of the dataset; the validation dataset, which composes 10% of the dataset; and this testing dataset, which makes up 20% of the dataset and is used for assessing how well the model works with fresh data. By stowing each node in a particular dataset, the training model may be applied to all nodes that have been previously specified for handling. The model may be trained independently using a different process that focuses on datasets created using a different model. Effective communication promotes methods that deliver hastily verified dataset information to multiple edge hubs in order to create a single model in advance of propagation [33]. At any point during a patient's assessment, a hub has started an undertone; it obtains the patient information that is a vector of shape. The data are taken into account as a model input while making a forward pass to the DQN for the binary output, i.e., either 1 or 0, which signals the patient either has a heart disease or does not, respectively. When making a diagnosis, we employ the bagging process, an ensemble method that combines this output of various models in order to generate highly accurate results. The worker receives the data, which is then forwarded to various worker nodes. Every worker adds it to their line, and the projected outcomes of every worker node are transmitted back to the worker who was authorized for this particular assignment. This primary element of the expectation class is then sent from bagging to the gateway's devices. When the desired results are inertia roots, this model gives the user the ability to disable this element. We demonstrated the deep learning model's performance as well as the ensemble method. It leads to an improved performance, quick responses, and reduced network overhead. The flow chart for the adopted approach has been presented in Figure 3.

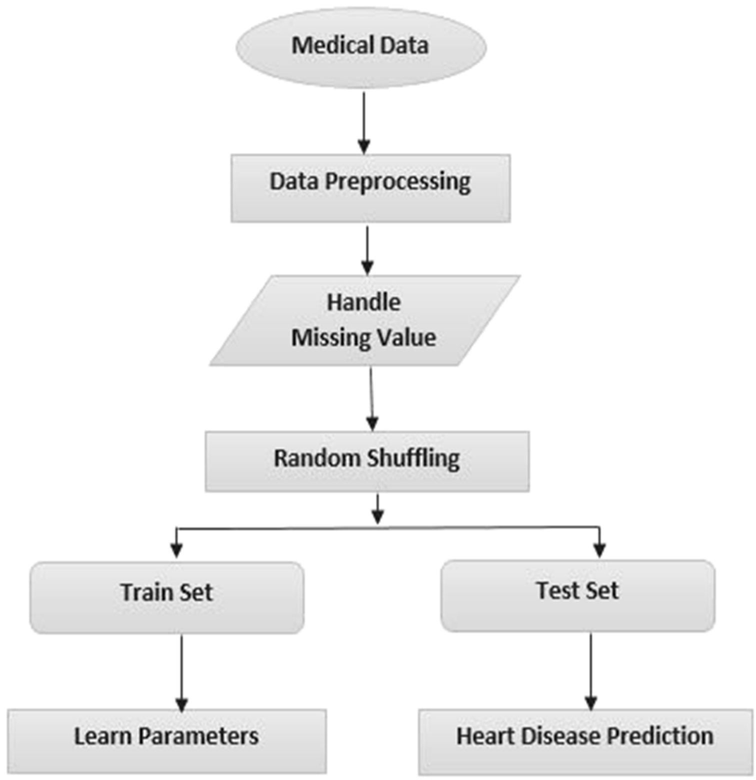

Figure 3. Flow chart of proposed method.

5.3. Proposed Deep Q Learning Based Prediction

The environment is taken to be in the state $s(t) \in S$ for some given time *t*, where *S* is the set of all conceivable states. The agent observes the current state s(t) and chooses the best course of action $a(t) \in A$, where *A* is the set of all of the potential action vectors. When action *a(t)* is taken, the environment changes from state *s(t)* to state *s(t + 1)*, and the agent receives the reward *r(t)*. The agent receives a reward for each state transition as it repeatedly acts in accordance with the policy and the observed state. As the reward that the agent receives in the future is considerably less significant for the decision-making at hand, the discounted reward can be obtained as:

$$D(t) = \sum_{k=0}^{\infty} \gamma^k r(t+k+1) \tag{1}$$

where $\gamma \in [0,1]$ represents the discount factor.

The agent's goal is to identify the best course of action that will maximize the long-term predicted discounted payoff. In accordance with a policy, when the environment is in states, the expected value of the discounted reward that the agent receives after taking action is provided by:

$$Q^{\pi}\big(s(t), a(t)\big) = E_{\pi}[D(t)|s(t) = S, a(t) = a] \tag{2}$$

In Equation (2), the expectation operation is represented as E_{π}.

Using Bellman's equation, *Q*-learning makes it possible to learn the *Q* function for a particular policy:

$$\begin{gathered} Q^{\pi}\big(s(t), a(t)\big) = E_{\pi}\big[r(t) + \gamma \textstyle\sum_{s(t+1)\in S} Pr\big(s(t+1)|s(t), a(t)\big) \times \\ \textstyle\sum_{a(t+1)\in A} \pi\big(s(t+1), a(t+1)\big) Q^{\pi}\big(s(t+1), a(t+1)\big)\big] \end{gathered} \tag{3}$$

In the above Equation (3), $Pr\big(s(t+1)|s(t), a(t)\big)$ denotes the state transition probability having action $a(t)$.

The Bellman optimality equation yields the optimal action-value function for the optimal policy π^*:

$$Q^*\big(s(t),a(t)\big) = r(t) + \gamma \max_{a(t+1)} Q^*\big(s(t+1),a(t+1)\big) \tag{4}$$

The state value function is obtained as:

$$V\big(s(t)\big) = \max_{a(t)\in A} Q\big(s(t),a(t)\big) \tag{5}$$

Employing $r(t)$ as the immediate reward and the state value in Equation (5), the *Q*-value can be updated as:

$$Q_{t+1}\big(s(t),a(t)\big) = (1-\alpha_t)Q_t\big(s(t),a(t)\big) + \alpha_t\Big(r(t) + \gamma V_t\big(s(t+1)\big)\Big) \tag{6}$$

where $\alpha_t \in (0,1]$ is the rate of learning.

The best *Q* policy is provided as, once the optimal *Q* function has been determined.

$$\pi^*(s,a) = arg \max_{a\in A} Q^*(s,a) \tag{7}$$

The agent chooses the action with the highest *Q* value for a given state when the function *Q(s,a)* has the form of a table. Algorithm 1 describes the proposed DQN algorithm and computation steps.

Algorithm 1 Proposed algorithm

function DQN (S, A, Pr, R, δ)

Input: S, A, Pr, R, δ

Output: $V_\pi(s), Q_\pi(s,a)$

Initialize:

S ←Set of states $S = \{s_1, s_2, \cdots s_n\}$

A ←Set of actions $A = \{a_1, a_2, \cdots a_n\}$

Pr ←Probability of transitioning the present state

R ←Set of rewards

γ ←Reward discount factor

Compute:

$\pi(\alpha|s) = p(A_k|S_k)$

for φ **do**

if $\varphi \sim \pi$ **and** $A_k \sim \pi(.|S_k)$ **do**

$$V_\pi(s) = E_{\varphi\sim\pi}[R(\varphi)|S_0]$$

$$V_\pi(s) = E_{A_k\sim\pi(.|S_k)}\left[\sum_{k=0}^{\infty} \delta^k R(S_k, A_k)|S_0\right]$$

$$Q_\pi(s,a) = E_{\varphi\sim\pi}[R(\varphi)|S_0, A_0]$$

$$Q_\pi(s,a) = E_{A_k\sim\pi(.|S_k)}\left[\sum_{k=0}^{\infty} \gamma^k R(S_k, A_k)|S_0, A_0\right]$$

end if

end for

return $V_\pi(s), Q_\pi(s,a)$

Exit

6. Experimental Setup

6.1. Dataset

The UCI database was deployed for the cardiac arrhythmia data mining, as has already been indicated. There are 452 examples of ECG signals in this database from people of various ages and sexes. From these signals, 279 features in total were identified. Among the most significant are: • Age (years) (years) • Sex (male = 0; female = 1) • Weight •

Height (cm) (kg) • QRS duration (average QRS length in milliseconds) • P-R distance (Average time interval between the start of waves P and Q in milliseconds) Q-T distance (Average time interval between the start of wave Q and end of wave T in milliseconds) • T Distance (Average time interval of wave T in milliseconds)• P Distance (Average P wave distance in milliseconds) • QRS(Degree vector angles on the screen) (Degree vector angles on the screen) T (screen-based degree vector angles) P (Degree vector angles on the screen) • QRST (Degree vector angles on the screen) (Degree vector angles on the screen) • J (Degree vector angles on the screen) (Degree vector angles on the screen) • Pulse rate (Heart rate per minute).

6.2. Performance Metrics

The important QoS factors that are used to evaluate the fog services are as follows:

1. Efficiency: as these fog nodes are nearer to the end user, it is closely integrated with the individual needs, which improves the performance and the efficiency of the entire framework. Combining the computational and storage resources across end-user devices and the cloud can also increase the performance [37,38].
2. Latency: certain essential services should not ever be suspended or deferred. Therefore, real-time stream processing for latency-sensitive applications, such as complicated event processing or stream mining, should be provided via fog computing to reduce the delay [39].
3. Reliability: to decrease latency, the fog computing technique should be employed to provide real-time stream processing for latency-sensitive applications such as complicated event processing or stream mining [39]. A fog computing-based system should be dependable in that it can carry out its assigned tasks and deliver the correct results under predetermined conditions and within a specified time limit.
4. Energy utilization: in a fog computing environment, resources may use some energy to deliver the required services or to resend some requests to the cloud for additional data processing. Users ought to take into account this energy consumption [40,41].
5. Scalability: in the midst of increasing operational demands, such as a rise in the volume of service requests or the application of resources, a scalable system can maintain or improve its level of effectiveness and performance.
6. Security: protecting the available cloud and fog data from threats using safe solutions is the major security problem in fog computing, along with device authentication at any gateway. An intrusion detection system (IDS) needs to be installed at each tier of the platform in order to address this issue [40].
7. Resource utilization: this describes the most effective use of a system’s resources and is crucial for maintaining efficiency.
8. Accuracy: this refers to any parameter that is close to the ideal value or the accepted benchmark. Each calculation must be completed accurately, and the output must be error-free.
9. Precision/Recall: two significant criteria for evaluating models and algorithms are recall and precision. The first displays the proportion of all relevant results that the algorithm successfully classified, while the second displays the proportion of algorithmic findings that are connected to the chosen topic.
10. Throughput: the term “throughput” specifies the greatest amount of data that can be sent from one point to another, or the most requested service rate that can be handled by the system in a specific amount of time.
11. Response time: the time it takes for a system query to be replied to after being sent. For successful computing, quick response times could be essential.
12. Execution time: this is the amount of time that passes between when a program begins to run and when the user or operating system terminates it.

6.3. Comparative Analysis

For the performance analysis, we compared our proposed approach with two similar approaches that were designed in the cloud platform. A cloud-IoT system based on the agents for remote heart rate monitoring was developed by the researchers in [36], which makes it easier to track and monitor patients with cardiovascular illness anywhere. The patient's heart rate can be recorded, located, stored, and analyzed using the suggested system. In an emergency situation, it may also be necessary to make a speedy choice. Any healthcare professional may use it, not just the hospital. Using five well-known supervised learning-based machine learning techniques, a cloud-based architecture for the early identification of cardiac disease was planned in [37].

Figure 4 illustrates the latency analysis of our proposed approach and its comparison with other similar approaches. For this evaluation, we have taken two related approaches in [36] and [37] and it can be established that our proposed model shows comparatively less latency than the Cloud-IoT HMS and cloud based Machine Learning approach.

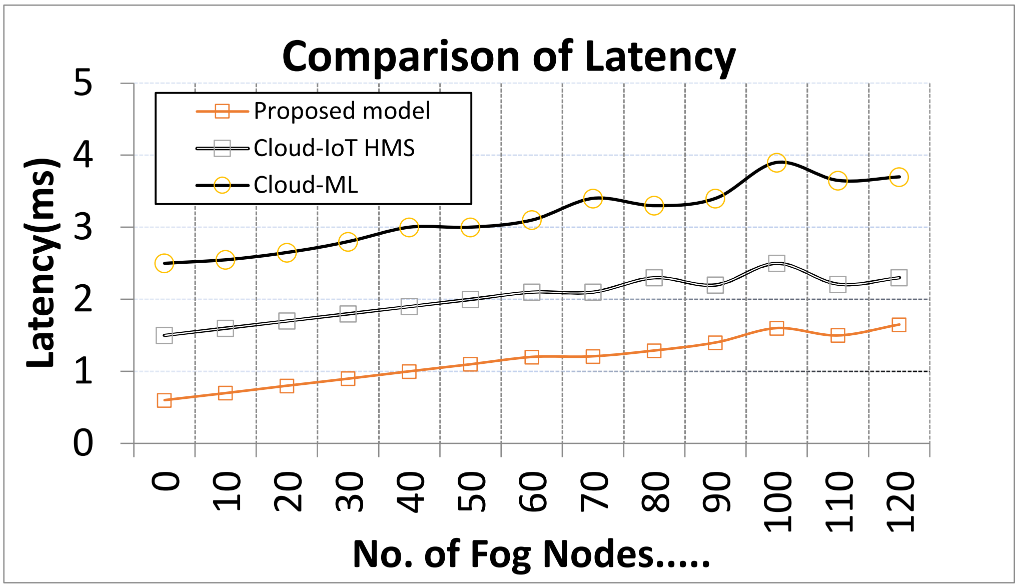

Figure 4. Comparison of Latency with Cloud-IoT HMS and Cloud-ML.

Figure 5 illustrates the execution time analysis of our proposed approach and its comparison with other similar approaches. For this evaluation, we have taken two related approaches in [36,37] and it can be found that our proposed model shows reasonably less execution time than the Cloud-IoT HMS and cloud based Machine Learning approach.

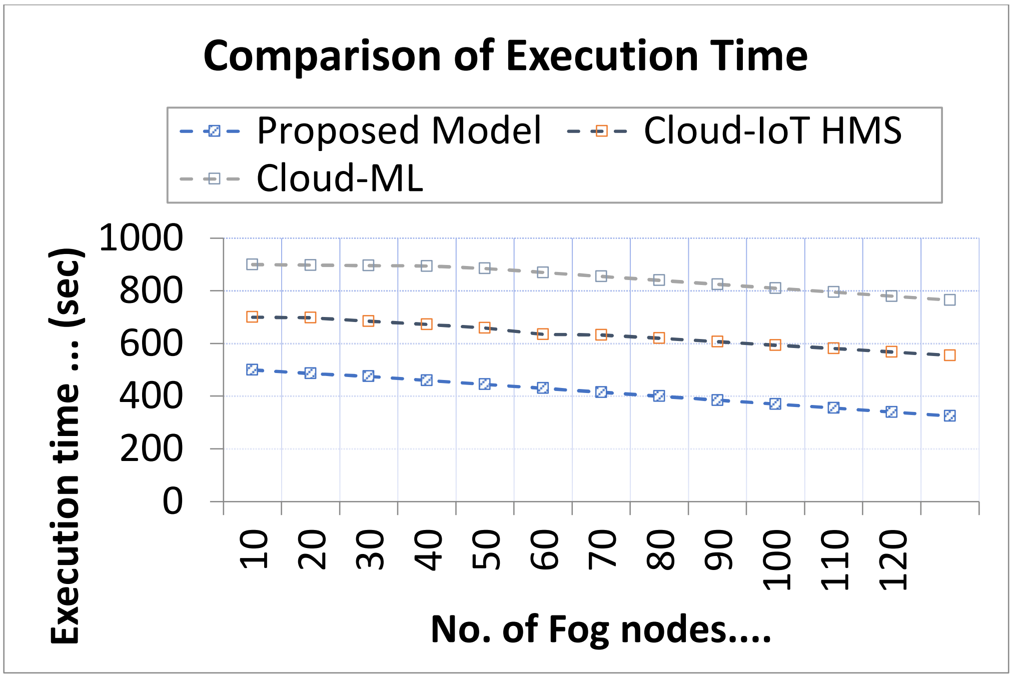

Figure 5. Comparison of Execution Time with Cloud-IoT HMS and Cloud-ML.

Figure 6 demonstrates the accuracy rate analysis of our proposed approach and its comparison with other analogous approaches. For this evaluation, we have taken two interrelated approaches in [36,37] and it can be found that our proposed model shows moderately high accuracy rate than the Cloud-IoT HMS and cloud based Machine Learning approach.

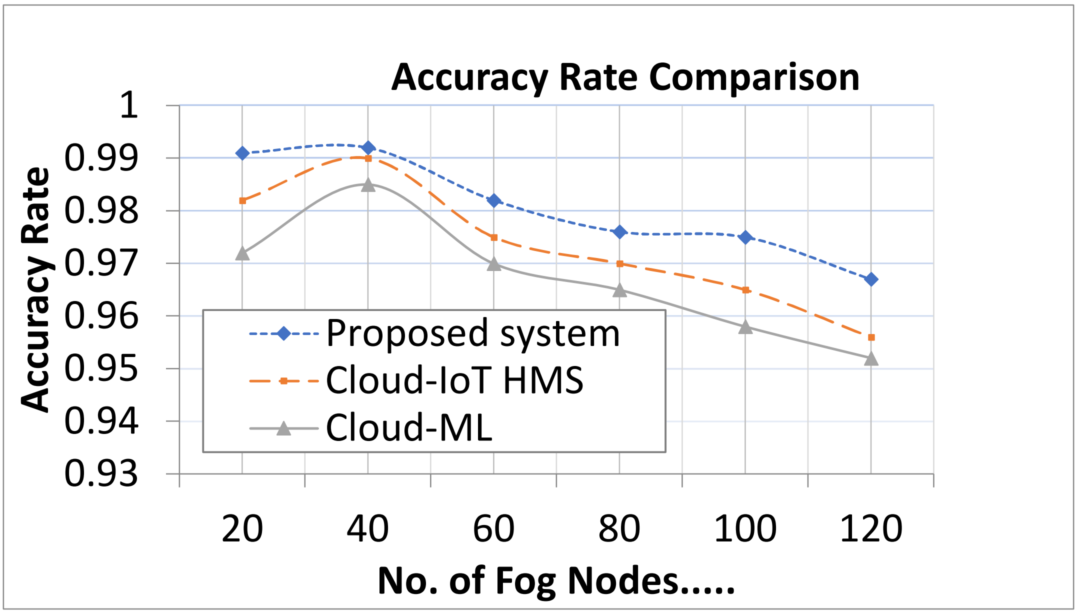

Figure 6. Comparison of Accuracy rate with Cloud-IoT HMS and Cloud-ML.

Figure 7 displays the energy expenditure investigation of our proposed approach and its comparison with other similar approaches. For this evaluation, we have taken two related approaches in [36,37] and it can be found that our proposed model consumes comparatively lesser energy than the Cloud-IoT HMS and cloud based Machine Learning system.

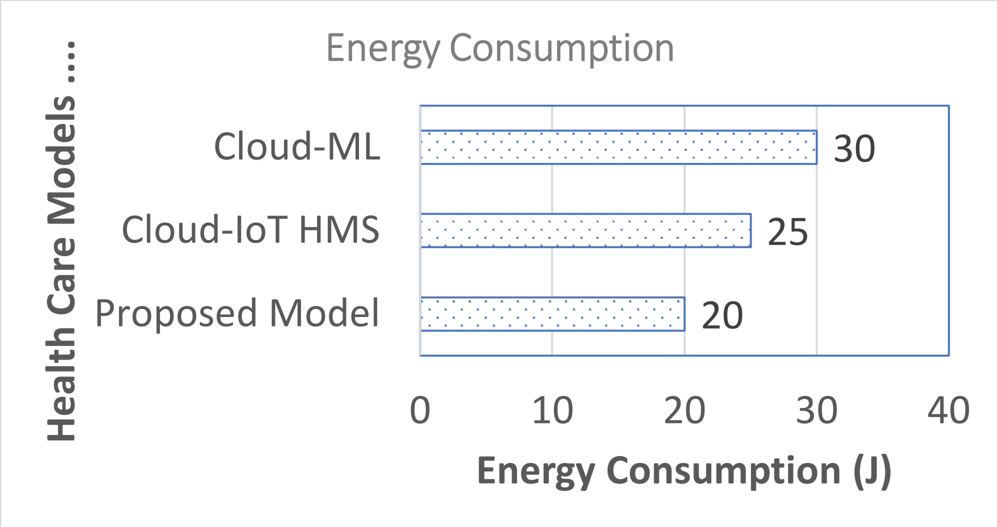

Figure 7. Comparison of Energy Consumption with Cloud-IoT HMS and Cloud-ML.

Figure 8 depicts the throughput testing of our proposed approach and its assessment with other similar approaches. For this assessment, we have taken two related approaches in [36,37] and it can be found that our proposed model consumes comparatively advanced throughput than the Cloud-IoT HMS and cloud based Machine Learning model.

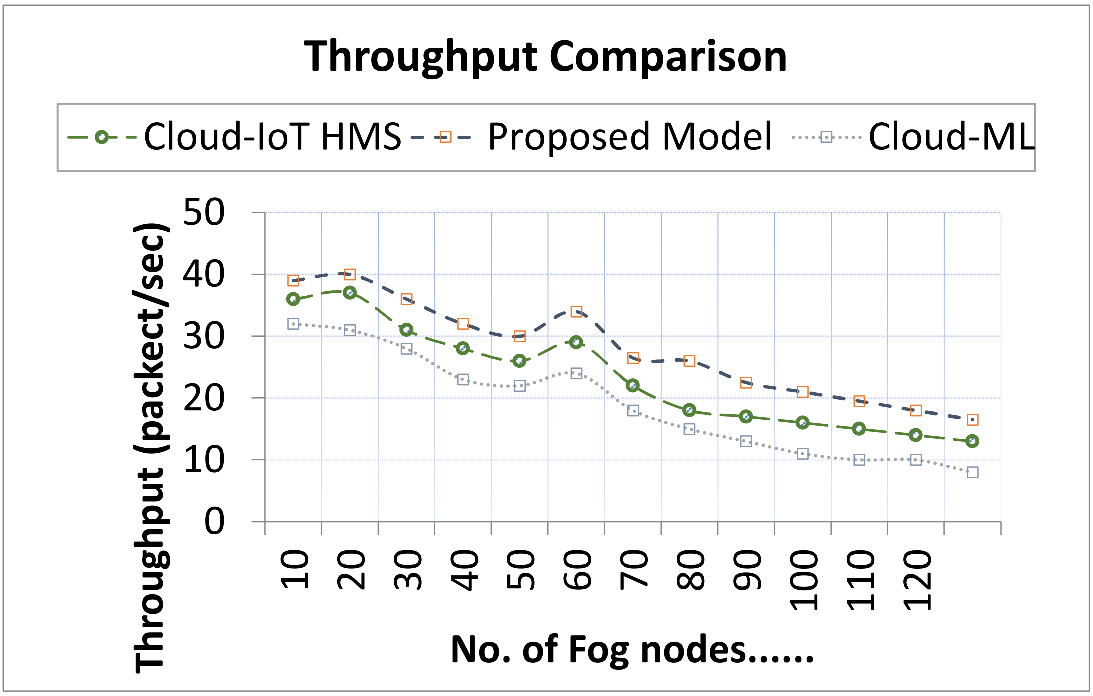

Figure 8. Comparison of Throughput with Cloud-IoT HMS and Cloud-ML.

7. Challenges and Future Work

In this section, we address the challenges that have the potential for further research to optimize and improve the proposed system in the future. Internet: as the proposed system is based on remote communication, as was discussed in the preceding sections, having a fast, reliable internet connection is crucial for a system that can monitor patients in actual environments. For high-risk patients who require ongoing monitoring, the system feedback will be unfavorable if the internet connection is slow or down. Implementing the redundancy technique in the network is one of the options that can be suggested here. Storage: when the system is constantly receiving and saving patient data, a substantial quantity of that data will end up in the memory of the device. One strategy to prevent running out of storage capacity is to only keep data on the fog layer for a short time—say, one day—and to keep the patient's data permanently on the cloud layer. Wearable devices: the elderly are one of our target community's objectives when implementing the suggested system. As was previously indicated, a patient must be connected to a number of probes in order to receive electrocardiography; nevertheless, the patient may not be able to complete this task on their own. As a result, the above system should be created to present the fewest difficulties to this kind of patient. Utilizing wearable technology is one of the recommendations. In this work, a useful technique for keeping track of patients' health was proposed. For patients with arrhythmias, this solution is built on fog computing and data mining. In order to reduce data transfer delays, patient information is used with fog technology rather than being sent to the cloud.

Author Contributions: Writing—original draft, S.S.T., M.R., N.T. and S.B.; Writing—review & editing, D.S.R. and J.S.A.F. All authors have read and agreed to the published version of the manuscript.

Funding: The research received no funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data that supports the findings of this paper is available upon reasonable request from the corresponding author.

Acknowledgments: The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University, for supporting this work through Small Groups Project under grant number RGP.1/142/43.

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

References

1. Mahmud, R.; Koch, F.L.; Buyya, R. Cloud-Fog Interoperability in IoT-enabled Healthcare Solutions. In Proceedings of the 19th International Conference on Distributed Computing and Networking (ICDCN '18), Varanasi, India, 4–7 January 2018; Article 32, pp. 1–10. https://doi.org/10.1145/3154273.3154347.
2. Islam, S.M.R.; Kwak, D.; Kabir, M.D.H.; Hossain, M.; Kwak, K.S. The internet of things for health care: A comprehensive survey. *IEEE Access* **2015**, *3*, 678–708.
3. 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.
4. Goyal, A.; Kahlon, P.; Jain, D.; Soni, R.K.; Gulati, R.; Chhabra, S.T.; Aslam, N.; Mohan, B.; Anand, I.; Patel, V.; et al. Trend in prevalence of coronary artery disease and risk factors over two decades in rural Punjab. *Heart Asia* **2017**, *9*, e010938.
5. Tuli, Shreshth, Nipam Basumataryand Rajkumar Buyya, Edgelens: Deep learning based object detection in integrated iot, fog and cloud computing environments. In Proceedings of the 4th IEEE International Conference on Information Systems and Computer Networks (ISCON 2019), Mathura, India, 21–22 November 2019.
6. Mutlag, A.A.; Ghani, M.K.A.; Arunkumar, N.; Mohammed, M.A.; Mohd, O. Enabling technologies for fog computing in healthcare IoT systems. *Future Gener. Comput. Syst.* **2019**, *90*, 62–78.
7. Faust, O.; Hagiwara, Y.; Hong, T.J.; Lih, O.S.; Acharya, U.R. Deep learning for healthcare applications based on physiological signals: A review. *Comput. Methods Programs Biomed.* **2018**, *161*, 1–13.
8. 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. https://doi.org/10.1016/j.future.2019.10.043.
9. Zhao, X.; Yang, K.; Chen, Q.; Peng, D.; Jiang, H.; Xu, X.; Shuang, X. Deep learning based mobile data offloading in mobile edge computing systems. *Future Gener. Comput. Syst.* **2019**, *99*, 346–355.
10. Lim, T.-S.; Loh, W.-Y.; Shih, Y.-S. A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. *Mach. Learn.* **2000**, *40*, 203–228.
11. Gill, S.S.; Arya, R.C.; Wander, G.S.; Buyya, R. 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*; Springer: Cham, Switzerland, 2018; pp. 1376–1383.
12. Juarez-Orozco, L.E.; Martinez-Manzanera, O.; Van Der Zant, F.M.; Knol, R.J.J.; Knuuti, J. 241 Deep learning in quantitative PET myocardial perfusion imaging to predict adverse cardiovascular events. *Eur. Heart J. Cardiovasc. Imaging* **2019**, *20*, jez145.005.

13. Acharya, U.R.; Fujita, H.; Oh, S.L.; Hagiwara, Y.; Tan, J.H.; Adam, M. Application of deep convolutional neural network for automated detection of myocardial infarction using ECG signals. *Inf. Sci.* **2017**, *415*, 190–198.
14. Azimi, I.; Takalo-Mattila, J.; Anzanpour, A.; Rahmani, A.M.; Soininen, J.-P.; Liljeberg, P. 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 (CHASE), Washington, DC, USA, 26–28 September 2018; pp. 63–68.
15. Gia, T.N.; Thanigaivelan, N.K.; Rahmani, A.M.; Westerlund, T.; Liljeberg, P.; Tenhunen, H. Customizing 6LoWPAN networks towards Internet-of-Things based ubiquitous healthcare systems. In Proceedings of the NORCHIP 2014—32nd NORCHIP Conference: The Nordic Microelectronics Event, Tampere, Finland, 27–28 October 2014; pp. 1–6.
16. Gómez, J.; Oviedo, B.; Zhuma, E. Patient monitoring system based on internet of things. *Procedia Comput. Sci.* **2016**, *83*, 90–97, 2016.
17. Fanucci, L.; Saponara, S.; Bacchillone, T.; Donati, M.; Barba, T.; Sanchez-Tato, I.; Carmona, C. Sensing devices and sensor signal processing for remote monitoring of vital signs in chf patients. *IEEE Trans. Instrum. Meas.* **2013**, *62*, 553–569.
18. Natarajan, K.; Prasath, B.; Kokila, P. Smart health care system using internet of things. *J. Netw. Commun. Emerg. Technol.* **2016**, *6, 37-42*.
19. Abdelmoneem, R.M.; Benslimane, A.; Shaaban, E.; Abdelhamid, S.; Ghoneim, S. A Cloud-Fog Based Architecture for IoT Applications Dedicated to Healthcare. In Proceedings of the ICC 2019-2019 IEEE International Conference on Communications (ICC), Shanghai, China, 20–24 May 2019; pp. 1–6.
20. Rajkomar, A.; Oren, E.; Chen, K.; Dai, A.M.; Hajaj, N.; Hardt, M.; Liu, P.J.; Liu, X.; Marcus, J.; Sun, M.; et al. Scalable and accurate deep learning with electronic health records. *NPJ Digit. Med.* **2018**, *1*, 18.
21. Pham, M.; Mengistu, Y.; Do, H.; Sheng, W. Delivering home healthcare through a cloud-based smart home environment (CoSHE). *Future Gener. Comput. Syst.* **2018**, *81*, 129–140.
22. Alam, G.R.; Munir, S.; Uddin, Z.; Alam, M.S.; Dang, T.N.; Hong, C.S. Edge-of-things computing framework for cost-effective provisioning of healthcare data. *J. Parallel Distrib. Comput.* **2019**, *123*, 54–60.
23. Moosavi, S.R.; Gia, T.N.; Nigussie, E.; Rahmani, A.M.; Virtanen, S.; Tenhunen, H.; Isoaho, J. End-to end security scheme for mobility enabled healthcare Internet of Things. *Future Gener. Comput. Syst.* **2016**, *64*, 108–124.
24. Alazeb, A.; Panda, B.; Almakdi, S.; Alshehri, M. Data Integrity Preservation Schemes in Smart Healthcare Systems That Use Fog Computing Distribution. *Electronics* **2021**, *10*, 1314. https://doi.org/10.3390/electronics10111314.
25. Alesanco, A.; García, J. Clinical assessment of wireless ECG transmission in real-time cardiac telemonitoring. *IEEE Trans. Inf. Technol. Biomed.* **2010**, *14*, 1144–1152.
26. Fettweis, G.P. The tactile internet: Applications and challenges. *IEEE Veh. Technol. Mag.* **2014**, *9*, 64–70.
27. Kraemer, F.A.; Braten, A.E.; Tamkittikhun, N.; Palma, D. Fog computing in healthcare–a review and discussion. *IEEE Access* **2017**, *5*, 9206–9222.
28. Nath, S.B.; Gupta, H.; Chakraborty, S.; Ghosh, S.K. A survey of fog computing and communication: Current researches and future directions. *arXiv* **2018**, arXiv:1804.04365.
29. Steele, R.; Lo, A. Telehealth and ubiquitous computing for bandwidth-constrained rural and remote areas. *Pers. Ubiquit. Comput.* **2013**, *17*, 533–543.
30. Yi, S.; Hao, Z.; Qin, Z.; Li, Q. Fog computing: Platform and applications. In Proceedings of the 3rd IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb), Washington, DC, USA, 12-13 November 2015; pp. 73–78.
31. Yi, S.; Li, C.; Li, Q. A survey of fog computing: Concepts, applications and issues. In: Proceedings of the 2015 Workshop on Mobile Big Data, Hangzhou, China, 21 June 2015; pp. 37–42.
32. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
33. Ahmid, M.; Kazar, O. A Cloud-IoT Health Monitoring System Based on Smart Agent for Cardiovascular Patients. In Proceedings of the 2021 International Conference on Information Technology (ICIT), Amman, Jordan, 14–15 July 2021; pp. 1–6. https://doi.org/10.1109/ICIT52682.2021.9491113.
34. Ahmed, M.R.; Mahmud, S.M.H.; Hossin, M.A.; Jahan, H.; Noori, S.R.H. A Cloud Based Four-Tier Architecture for Early Detection of Heart Disease with Machine Learning Algorithms. In Proceedings of the 2018 IEEE 4th International Conference on Computer and Communications (ICCC), Chengdu, China, 7–10 December 2018; pp. 1951–1955. https://doi.org/10.1109/CompComm.2018.8781022.
35. Secinaro, S.; Calandra, D.; Secinaro, A.; Muthurangu, V.; Biancone, P. The role of artificial intelligence in healthcare: A structured literature review. *BMC Med. Inform. Decis. Mak.* **2021**, *21*, 125. https://doi.org/10.1186/s12911-021-01488-9.
36. Kar, U.K. The Future of Health and Healthcare in a World of Artificial Intelligence. *Arch. Biomed. Eng. Biotechnol*. **2018,** *1, 1-7*. https://doi.org/10.33552/abeb.2018.01.000503.
37. Rodriguez-Romero, V.; Bergstrom, R.F.; Decker, B.S.; Lahu, G.; Vakilynejad, M.; Bies, R.R. Prediction of Nephropathy in Type 2 Diabetes: An Analysis of the ACCORD Trial Applying Machine Learning Techniques. *Clin. Transl. Sci*. **2019**, *12*, 519–528. https://doi.org/10.1111/cts.12647.
38. Kwak, G.H.; Hui, P. DeepHealth: Review and Challenges of Artificial Intelligence in Health Informatics, No. Ml, 2019. Available online: http://arxiv.org/abs/1909.00384. (accessed on 7 October 2022).

39. Yang, S.; Zhu, F.; Ling, X.; Liu, Q.; Zhao, P. Intelligent Health Care: Applications of Deep Learning in Computational Medicine. *Front. Genet*. **2021**, *12*, 607471. https://doi.org/10.3389/fgene.2021.607471.
40. Kaul, D.; Raju, H.; Tripathy, B.K. Deep Learning in Healthcare. *Stud. Big Data* **2022**, *91*, 97–115. https://doi.org/10.1007/978-3-030-75855-4_6.
41. Esteva, A.; Robicquet, A.; Ramsundar, B.; Kuleshov, V.; Depristo, M.; Chou, K.; Cui, C.; Corrado, G.; Thrun, S.; Dean, J. A guide to deep learning in healthcare. *Nat. Med*. **2019**, *25*, 24–29. https://doi.org/10.1038/s41591-018-0316-z.