IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 26 March 2023, accepted 3 April 2023, date of publication 11 April 2023, date of current version 4 May 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3266374

RESEARCH ARTICLE

DeepMist: Toward Deep Learning Assisted Mist Computing Framework for Managing Healthcare Big Data

SUJIT BEBORTTA[1], (Student Member, IEEE), SUBHRANSHU SEKHAR TRIPATHY[2], SHAKILA BASHEER[3], AND CHIRANJI LAL CHOWDHARY[4]

[1]Department of Computer Science, Ravenshaw University, Cuttack, Odisha 753003, India
[2]Department of Computer Science and Engineering, Dhaneswar Rath Institute of Engineering and Management Studies (DRIEMS), Autonomous College, Cuttack, Odisha 754025, India
[3]Department of Information Systems, College of Computer and Information Science, Princess Nourah bint Abdulrahman University, Riyadh 11671, Saudi Arabia
[4]School of Information Technology and Engineering, Vellore Institute of Technology, Vellore 632014, India

Corresponding author: Chiranji Lal Chowdhary (prof.chowdhary@gmail.com)

This research is supported by Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2023R195) Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

ABSTRACT The prevalence of heart disease has remained a major cause of mortalities across the world and has been challenging for healthcare providers to detect early symptoms of cardiac patients. To this end, several conventional machine learning models have gained popularity in providing precise prediction of heart diseases by taking into account the underlying conditions of patients. The drawbacks associated with these methods are a lack of generalization and the convergence rate of these methods being much slower. As the healthcare data associated with these systems scale up leading to healthcare big data issues, a Cloud-Fog computing-based paradigm is necessary to facilitate low-latency and energy-efficient computation of the healthcare data. In this paper, a DeepMist framework is suggested which exploits Deep Learning models operating over Mist Computing infrastructure to leverage fast predictive convergence, low-latency, and energy efficiency for smart healthcare systems. We exploit the Deep Q Network (DQN) algorithm for building the prediction model for identifying heart diseases over the Mist computing layer. Different performance evaluation metrics, like precision, recall, f-measure, accuracy, energy consumption, and delay, are used to assess the proposed DeepMist framework. It provided an overall prediction accuracy of 97.6714% and loss value of 0.3841, along with energy consumption and delay of 32.1002 mJ and 2.8002 ms respectively. To validate the efficacy of DeepMist, we compare its outcomes over the heart disease dataset in convergence with other benchmark models like Q-Reinforcement Learning (QRL) and Deep Reinforcement Learning (DRL) algorithms and observe that the proposed scheme outperforms all others.

INDEX TERMS Deep learning, mist computing, heart disease prediction, performance evaluation, latency, energy efficiency.

I. INTRODUCTION

The development of cutting-edge means of communication and storage, the medical, agricultural, and other fields have significantly benefited from incorporating smart gadgets into everyday life [1]. Because of the tremendous progress made in digital technology over the past few years, the Internet of Things (IoT) has had a significant influence on our everyday life. The IoT is often conceptualized as a network of interconnected computing systems that share information to achieve specific goals. These systems often include sensors, actuators, and processors [2]. IoT-enabled devices employ sensing technologies to produce extensive data, which is then transferred via fog computing or cloud computing to locations where deep learning algorithms can be used to make decisions accurately. Fog computing, which includes

The associate editor coordinating the review of this manuscript and approving it for publication was Ali Kashif Bashir.

cloud computing standards, has become the foundation of an advanced economy relying on the Internet to provide customer services [3]. However, owing to the significant setback to speed of reaction, cloud computing is unsuitable for applications that require regular feedback. Big data management in the context of this IoT, fog, and cloud computing have all risen to prominence due to their user-friendliness and ability to provide reaction features dependent on the tracked target applications. These new technologies give edge devices the ability to store, compute, and communicate with each other. This improves mobility issues, security, and privacy along with optimizing latency and network bandwidth so that fog computing can work well with real-time or latency-sensitive applications [4], [5].

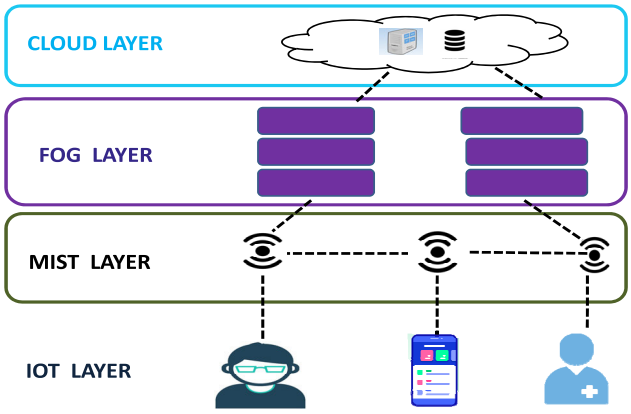

FIGURE 1. An overview of IoT-based health monitoring system with mist, fog, and cloud computing layers.

Instead of merely making wearable devices, it's important to build a full community wherein sensors in a body-area network may link up data to cloud storage with the help of this IoT network [6], [7], [8]. Figure 1 depicts the architectural components often required in IoT systems for healthcare (Health-IoT). The three key parts of the architecture are the body area sensor networks, the Internet-connected access points, and the edge as well as cloud computing components. Through this system, several applications offer services to various system stakeholders. Caregivers, family members, and other approved parties have access to data generated by sensors attached to users, allowing them to monitor the subject's vital signs whenever and wherever they choose [5].

Recently, deep learning [9] has been demonstrated to be effective in mixed-modality data settings, natural language processing, and sequence prediction, adding to its already widespread use in areas such as computer vision [10] and speech synthesis. Additionally, ensemble learning [9] is utilized to combine the strengths of various classifiers. Classifiers can be improved by ensemble learning [11]. The bagging classifier ensemble method uses randomized data samples to learn the base classifier, and then uses majority or average polling to reach a consensus on the best course of action. Compared to using a single estimator, variance is reduced when datasets are distributed randomly. Deep learning can accurately forecast and classify healthcare data [9].

In the present work, we offer DeepMist framework, which is a computationally-aware model to facilitate comprehensive identification and automated diagnosis of cardiac illness by incorporating IoT paradigms and Deep Q Network (DQN) algorithm. DeepMist administers medical care over light-weight mist computing nodes in the IoT physical layer and connects to the fog layer for dissemination of information about cardiac patients intelligently captured from other IoT-enabled devices. By exemplifying the use of the iFogSim paradigm [12], as well as demonstrating the use, together with design to implement the mist features to achieve the objectives, DeepMist offers the following kind of support:

- Develop a deep learning ensemble method that works with a broad framework architecture to promote mist-fog processing of acquired data.
- Using the deep learning ensemble technique DeepMist, a framework for lightweight computing of cardiac patient diagnosis was built over the mist computing layer. Relevant data processing is modelled with the help of a system that integrates the IoT, mist computing, fog computing environment, and the cloud.
- This efficacy of DeepMist deployment is done through various performance indicators, such as precision, recall, f-measure, accuracy, energy efficiency, and transmission delay.
- The fundamental objective is to reduce the computational load over resource constrained IoT devices and offload the computationally intensive workloads to the mist computing devices for timely dissemination of critical healthcare information.

The research is segmented into 6 sections, and section II reviews previous studies that are pertinent to the topic at hand. Section III emphasizes the work's technical significance and illustrates the DeepMist framework's workflow. The system infrastructure for the DeepMist model is covered in section IV. Section V summarizes the model's essential implementation modules and simulation outcomes along with the performance evaluation strategies adopted in this work. Section VI concludes the experimental outcomes of DeepMist framework and discusses future studies.

II. RELATED WORK

The related work section of the research article likely discusses prior studies that have explored the use of mist computing and deep learning in healthcare or associated domains. It may also discuss existing healthcare big data management frameworks and their limitations, highlighting the need for a mist computing framework that leverages deep learning to address these limitations. In recent years, there has been an increase in the use of deep learning techniques to analyze healthcare big data, such as electronic health records, medical images, and sensor data from wearable devices. An emerging field called ''mist computing'' has shown great promise for

processing data at the network's edge, closer to where the data is generated, to reduce latency and bandwidth requirements and improve privacy and security. The application of deep learning to healthcare data management is an emerging topic of study, and there have been several publications in this area. It presents a framework that employs a mist computing strategy to manage extensive healthcare data using deep learning techniques. Mist computing is a distributed computing model that combines cloud computing and edge computing to process data at the network's edge in a resource-constrained environment near where the data is generated.

Fog computing provides a fundamental paradigm for efficiently processing healthcare data in the medical area, which may be retrieved from a range of IoT-empowered devices. Due to the proximity of fog computing enabled devices to IoT-enabled devices, much lower latency, delay, or reaction time can be achieved when using these devices to handle cardiac patients' data as opposed to cloud-based datacentres.

There have been efforts to combine deep learning and fog computing to manage big data in healthcare. For instance, [12] Journal of Medical Systems study proposed a deep learning-based framework for analyzing EEG data at the network's edge. The authors processed the EEG data in real-time using mist computing and a deep learning model to classify the EEG signals and detect epileptic seizures. In [13] study proposed a mist computing framework for the remote monitoring of Parkinson's disease patients. The framework employed wearable sensors to gather data on patients' movements and a deep learning model to analyze the data and detect motor symptoms of the disease. In addition, Ali and Ghazal [14] present an IoT-based platform for e-health monitoring services. This architecture is predicated on a software-defined network (SDN) that can gather data employing a smart phone and by leveraging voice control system, to improve the patient's wellbeing.

Rajasekaran et al. [15], proposed an Autonomous-Monitoring-System (AMS) paradigm for IoMT applications (IoMT). This model has the potential to serve the healthcare industry. This study utilizes a reward scheme and exploits the Analytics-Hierarchy-Process (AHP) to distribute resources fairly among the many nodes of the model under consideration. In terms of energy consumption, a cloud-based simulation and testing framework has demonstrated that the autonomous monitoring system is preferable to the FGCS technique. Even still, the lag time between terminals depicts the long latency time of patient planning. The SMART-For-Gateway (SFG) concept described by Constant et al. [16] offers a prospective filter, towards developing an intelligent filtering system, in-depth analysis, and targeted data transfers between IoT-enabled devices (wearable) and a data modeling system. While the suggested methodology streamlines the display of system execution time and energy usage, it ignores critical performance characteristics such as latency. For patent records of FHIR-based electronic health data, Rajkumar et al. [17], suggested an approach based on deep learning framework with great scalability and precision. Accurate analysis of several clinical prospects from various perspectives is made possible by the suggested model's usage of FHIR delineation, which operates over a strategic environment by leveraging deep learning algorithm without the need to harmonize site-specific data.

The work proposed by Moosavi et al. [18] suggested the use of an end-to-end handshake protocol for security analysis of IoT-enabled healthcare systems over datagram transport layer security (DTLS) by facilitating segregation without reintegrating the devices in the physical layer. Based on the findings, it was determined that the suggested strategy successfully cut transmission cost by 25% and latency by less than 16%. Despite the availability of an IoT-enabled framework, Azimi et al. [19] studied the efficiency of deep learning techniques based on Convolutional Neural Network (CNN) modeling in tandem with some classification-based approaches for investigation of Hierarchical-Edge-Based-Deep-Learning (HEDL) towards edge computing-enabled medical applications [20], [21]. Further, Adelmoneem et al. [22], proposed a task scheduling and allocation method to efficiently distribute healthcare jobs for processing healthcare data. The performance of CBFA is solely examined in terms of latency using the iFogSim simulator. Research created general healthcare applications on a modest scale, and the studies focused on something other than healthcare apps for diagnosing the health state of heart patients.

To develop deep learning tools and implement robotic monitoring, Verma et al., [21] propose a FETCH model that communicates with edge computing devices. It provides a useful framework for dealing with real-world healthcare issues including heart disease and others. FogBus is used by the developed Fog enabled cloud computing framework to present the values for metrics such as power consumption, jitter, network bandwidth, latency, accuracy and execution time. In Table 1, the comparison for the related studies is presented in contrast with the proposed DeepMist framework.

To realize the full potential of the IoT-based mist computing for healthcare systems, it is necessary to find solutions to the following difficulties [11], [12], [19], [21], [25]:

1. In order to effectively treat cardiac patients, we need a healthcare application built on the Internet of Things (IoT) that can handle massive amounts of data with minimal energy requirements and quick response times.
2. For client workloads to be executed in mist computing settings with maximum resource utilization and within their respective deadlines, a well-organized resource scheduling approach is essential.
3. Auto-assessing the severity of the cardiac disease calls for a mist computing model based on an ensemble of deep learning algorithms.

TABLE 1. Comparison table for proposed DeepMist model features with benchmark models and frameworks.

Proposed Models	Authored by	Features							
		Mist computing	Fog computing	Deep learning	IoT	Energy consumption	Training Accuracy	Testing Accuracy	Delay
AMS	Rajasekaran et al., [15]		✓			✓		✓	
EOTC	Alam et al.,[20]		✓		✓				
FETCH	Verma et al.,[21]		✓	✓	✓	✓	✓	✓	✓
FIH	Mahmud et al.,[12]		✓		✓	✓			✓
CFBA	Randa M et al., [22]		✓					✓	✓
DeepMist	Proposed Work	✓	✓	✓	✓	✓	✓	✓	✓

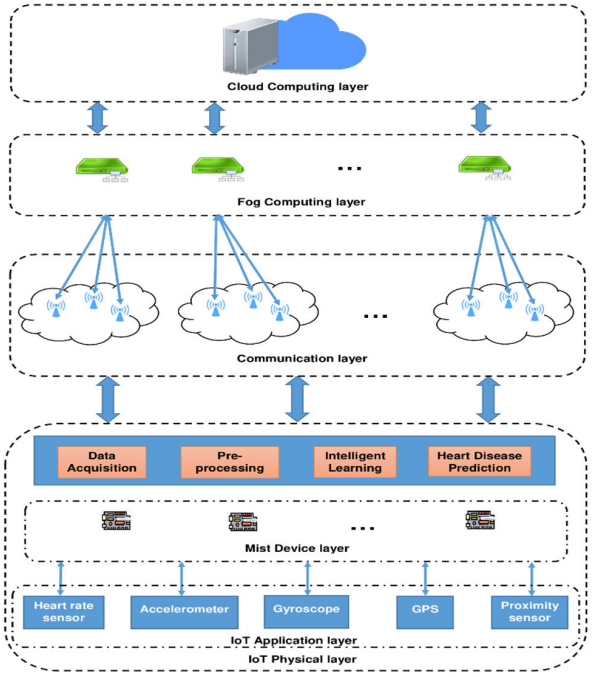

FIGURE 2. Proposed DeepMist Framework along with its functional modules.

III. PROPOSED FRAMEWORK

The proposed DeepMist framework is illustrated in this section along with an emphasis on the integration of IoT application layer, mist computing layer, fog layer and cloud layer [23], [24]. The DeepMist framework incorporates intelligence by exploiting the DQN algorithm to facilitate prediction of heart disease in a distributed manner over the mist computing devices. The proposed paradigm is four layered namely, the IoT physical layer, communication layer, fog computing layer, and cloud computing layer [25], [26], [27]. Figure 2 illustrates the proposed DeepMist framework along with all its functional modules.

We start with illustrating the functionality corresponding to individual layers in a bottom up approach. The IoT physical layer is placed at the bottom most layer and is further subdivided into three sub-layers viz., IoT application layer, the mist computing layer, and processing layer. The IoT application layer comprises of all the sensory devices like heart rate sensors, accelerometer, gyroscopes, GPS sensors for providing location information, and proximity sensors. The mist computing layer is in charge of maintaining the IoT devices and capturing situational data from them [23], [24], [28], [29]. The mist computing devices perform computations locally over the real-time sensor data or non-real-time data furnished by the end users via end devices like smart phones, laptops, or PCs. The DeepMist framework is incorporated at the mist computing layer to acquire bulk sensory data from a distributed network of IoT devices or end devices. The mist computing layer facilitates processing of the offloaded data from low power and resource constrained IoT devices and also reduces the amount of data transmitted over the communication network to the higher layers like fog and cloud computing layers. By exploiting the mist computing paradigm, the aforementioned IoT application layer devices can both pre-process and execute the IoT generated tasks locally with minimal latency and energy consumption [23]. The IoT physical layer facilitates processing of the sensor data and incorporates the pre-processing and intelligent prediction module.

The second layer of the DeepMist architecture i.e., the communication layer is responsible for connecting the IoT devices and mist computing devices to the higher layers like fog and cloud computing layers. The IoT devices entail restrained storage resources, bandwidth, and computing power and consequently require to be connected to its higher layers. This optimizes the network and computing resources as during situations when the computational load on the IoT devices increase, they can efficiently exploit the communication layer to offload their computational workloads to nearby devices or fog nodes [28], [29], [30].

The third layer in the DeepMist architecture is the fog computing layer and constitutes of the following components: fog nodes, gateways, and access networks. The gateways comprise of operating system and processing unit of this layer which may include hardware, microcontrollers, and signal processing unit. It provides the necessary infrastructure and resources to the fog nodes for execution of tasks generated by IoT physical layer. The access network is further responsible for specifying the form communication incorporated by mist devices to transmit data to the fog layer. Some examples of access network paradigms include WiFi, Ethernet, Wide Area Network (WAN), and Metropolitan Area Network (MAN) [28], [29], [30], [31]. The computational workloads offloaded from the IoT physical layer may be assigned to fog nodes over Wireless Local Area Network (WLAN) [28]. If a fog node becomes overloaded with computationally intensive workloads, the edge orchestrator is invoked which is responsible for distributing the workload among other fog nodes in this layer by implementing MAN connection [31], [32]. After performing the aforementioned processing of tasks, the data may be stored locally over the fog devices or may be further transmitted to the remote cloud servers via WAN connections depending upon the type of application and the requirements of the user [31].

IV. SYSTEM MODEL

We consider a Mist network (as depicted in Figure 2) to model the proposed DeepMist framework. The network comprises of a heterogeneous collection of IoT devices which push their data to the Mist computing nodes to facilitate offloading of computational workloads over resource constrained IoT devices [32]. We represent the set of mist nodes in the network as $M = \{M_1, M_2, \ldots, M_n\}$. The set of access points in the Mist network, denoted as A_m serve as the end devices to host the tasks being offloaded from the Mist layer to its consecutive Fog layer. We model the network as a collection of IoT devices, mist nodes, and fog nodes connected through some communication link by leveraging graph structure such that $G = (A_m \cup M \cup I, L)$ where I provides set of IoT devices and L can be denoted as the set of communication links between IoT devices, mist nodes, and access points. In our proposed framework, the mist nodes may operate using physical servers or through virtual machines. Further, to optimize the network cost [33], [34], [35], the mist nodes are considered to be less than the access points i.e., $|M| < |A_m|$. The number of computational tasks being offloaded from IoT device layer to mist nodes are depicted through the relationship $n_{tasks} := (p_{tasks}, s_{tasks})$ where p_{tasks} denote the CPU cycles required for processing the tasks and s_{tasks} represent size of healthcare tasks. Considering the number of mist nodes to be lower than the access points therefore, the tasks incur a multihop communication path.

A. MIST COMPUTING MODEL

The mist computing framework offers a low energy and latency-aware solution for processing of computationally intensive tasks generated by IoT devices and aids in improving Quality of Experience (QoE) for IoT users. The mist computing layer accomplishes task offloading in basically three steps.

1) The computational tasks c_{tasks} generated from wearable IoT devices I being transmitted to the mist nodes via wireless uplink paths l_{up}, where $l_{up} \in L$.
2) The c_{tasks} can be processed by the mist nodes having computing capacity $\rho^m_{I,L}$ where the allocation of mist nodes can be defined as $\varphi = \left\{\rho^m_{I,L} | \forall I, L \in G\right\}$, such that $\varphi_{\max}$ denotes the maximum computing capacity by a mist node.
3) The offloaded tasks c_{tasks} processed by the mist nodes are further transmitted to the IoT device layer through wireless downlink paths l_{down}, where $l_{down} \in L$.

B. DEEP REINFORCEMENT LEARNING MODEL

This study proposed the use of deep Q network (DQN) algorithm which, is a popularly used algorithm for achieving rapid convergence and high prediction accuracy [36]. The DQN algorithm was used for training the healthcare data considered in this study. The DQN algorithm finds its inception from the reinforcement learning model, which can be viewed as an interactive framework which allows a learning agent to adopt some random action and change its state based on the reward for the action in predominance of deterministic environment. Hence, the reinforcement learning can be described as a Markov Decision Process (MDP) [37], [38], [39].

1) VALUE FUNCTION ESTIMATION

We require to estimate the value function for determining the goodness of a particular action to be in a certain state. The notion of goodness of a specific action committed by the learning agent is determined on the basis of the number of expected future rewards and the precision in terms of expected number of returns [38]. The value function for actions may be described as a specific way of acting as a consequence of the learning process based on the policies.

For some state $\mathcal{s}$, considering the policy to be π, the value function can be denoted as $\upsilon_\pi(\mathcal{s})$ and can hence be stated as the expectation value of return $\mathbb{E}\left[\theta_t\right]$, operating from some state $\mathcal{s}$ and under policy π. Hence, we can obtain the formal definition of the value function for the MDPs as,

$$\upsilon_\pi(\mathcal{s}) = \mathbb{E}\left[\theta_t | \mathcal{S}_t = \mathcal{s}\right] \tag{1}$$

From the above Eq.(1), the state-value function is obtained, where $\mathbb{E}[.]$ represents the expected value for some random variable under policy π and time frame $\mathcal{t}$.

The return θ_t can be obtained for successive time frames of the learning system as,

$$\theta_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} \cdots + \gamma^i r_{t+j} \tag{2a}$$

$$\theta_t = r_{t+1} + \gamma\left(r_{t+2} + \gamma^1 r_{t+3} \cdots + \gamma^{i-1} r_{t+j}\right) \tag{2b}$$

$$\theta_t = r_{t+1} + \gamma\theta_{t+1} \tag{2c}$$

where r_{t+j} denotes the reward sequence such that $r_{t+j} \in R$, $j = 1, 2, 3, \ldots, m$ and γ^i denotes the discount parameter for $i = 0, 1, 2, \ldots, n$ subject to the condition $0 \leq \gamma \leq 1$.

Now, by using the expression for return obtained in Eq.(4), we can modify Eq.(1) by recursively solving for policy π and state $\mathcal{s}$, we obtain,

$$\upsilon_\pi(\mathcal{s}) = \mathbb{E}_\pi\left[r_{t+1} + \gamma\theta_{t+1}|\mathcal{S}_t = \mathcal{s}\right] \tag{3a}$$

$$\upsilon_\pi(\mathcal{s}) = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty}\gamma^k r_{t+k+1}|\mathcal{S}_t = \mathcal{s}\right] \tag{3b}$$

The above expression in Eq.(3b) is referred to as the equation for state value function υ_π.

Following the above convention for the state value function, the action value function for the above problem can be defined as the expected return or feedback from $\mathcal{s}$ adopting action a for policy π can be stated as,

$$u_\pi(\mathcal{s}, a) = \mathbb{E}_\pi\left[\theta_t|\mathcal{S}_t = \mathcal{s}, A_t = a\right] \tag{4}$$

Substituting the value in Eq.(2c) over the above equation we get,

$$u_\pi(\mathcal{s}, a) = \mathbb{E}_\pi\left[r_{t+1} + \gamma\theta_{t+1}|\mathcal{S}_t = \mathcal{s}, A_t = a\right] \tag{5a}$$

$$u_\pi(\mathcal{s}, a) = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty}\gamma^k r_{t+k+1}|\mathcal{S}_t = \mathcal{s}, A_t = a\right] \tag{5b}$$

We further estimate the Bellman's optimality equation to preserve self-consistency of state values in Eq.(3b) [39]. It makes intuitive sense that corresponding to an optimal policy, the value of a state must be equal to the average return pertaining to the best action undertaken by the learning agent from the state, and this can be defined as:

$$\begin{aligned}
\upsilon^*(\mathcal{s}) &= \max_{a\in A} u_\pi^*(\mathcal{s}, a) \\
&= \max_{a\in A} \mathbb{E}_\pi^*\left[\theta_t|\mathcal{S}_t = \mathcal{s}, A = a\right] \\
&= \max_{a\in A} \mathbb{E}_\pi^*\left[r_{t+1} + \gamma\theta_{t+1}|\mathcal{S}_t = \mathcal{s}, A_t = a\right] \\
&= \max_{a\in A} \mathbb{E}_\pi^*\left[r_{t+1} + \gamma\upsilon^*\left(\mathcal{S}_{t+1}\right)|\mathcal{S}_t = \mathcal{s}, A_t = a\right] \\
&= \max_{a\in A} \sum_{\mathcal{s}', r} p\left(\mathcal{s}', r|\mathcal{s}, a\right)\left[r + \gamma\upsilon^*\left(\mathcal{s}'\right)\right]
\end{aligned} \tag{6}$$

where $\mathcal{s}'$ represents the successive states such that $\mathcal{s} \in \mathcal{S}$, and $p\left(\mathcal{s}', r|\mathcal{s}, a\right)$ gives the probability for state $\mathcal{s}'$ and reward r conditioned over present state $\mathcal{s}$ and action a.

Now, the Bellman's optimality equation pertaining to the action value function in Eq.(4) can be obtained as [39],

$$u^*(\mathcal{s}, a) = \mathbb{E}_\pi\left[r_{t+1} + \gamma \max_{a'} u^*\left(\mathcal{S}_{t+1}, a'\right)|\mathcal{S}_t = \mathcal{s}, A_t = a\right] \tag{7a}$$

$$= \sum_{\mathcal{s}', r} p\left(\mathcal{s}', r|\mathcal{s}, a\right)\left[r + \gamma \max_{a'} u^*\left(\mathcal{s}', a'\right)\right] \tag{7b}$$

2) ACTION SELECTION STRATEGY

The state-action value of function $u_\pi(\mathcal{s}, a)$ can be enhanced in terms of the learning modules global search capability by adopting the e-greedy action selection method under the policy π and can be given as,

$$\pi(\mathcal{s}) = \begin{cases} \arg\max_a u_\pi(\mathcal{s}, a), & 0 \leq \varphi < e \\ \forall a \in A, & e \leq \varphi \leq 1 \end{cases} \tag{8}$$

It can be noted from the above formulation in Eq.(8), when $\varphi < e$, the action a corresponding to largest value for state-action function $u_\pi(\mathcal{s}, a)$ is selected. However, when $\varphi \geq e$, then a random action a is selected such that $a \in A$.

Hence, the Q-table can be updated for some temporal difference prediction as,

$$\begin{aligned}
u_\pi\left(\mathcal{s}_t, a_t\right) \leftarrow\ & u_\pi\left(\mathcal{s}_t, a_t\right) \\
&+ \sigma\left[\theta_t + \gamma \max_{a_{t+1}} u_\pi\left(\mathcal{s}_{t+1}, a_{t+1}\right)\right. \\
&\left. - u_\pi\left(\mathcal{s}_t, a_t\right)\right]
\end{aligned} \tag{9}$$

where:

- $\mathcal{s}_{t+1}$ = new state after action a_t is taken over current state $\mathcal{s}_t$.
- a_{t+1} = optimal action in state $\mathcal{s}_{t+1}$.
- $\theta_t + \gamma \max_{a_{t+1}} u_\pi\left(\mathcal{s}_{t+1}, a_{t+1}\right)$ = actual value of Q-function.
- $u_\pi\left(\mathcal{s}_t, a_t\right)$ = estimated value of Q-function.

C. DEEP Q NETWORK (DQN) ALGORITHM

In order to fit the value function $u_\pi(\mathcal{s}, a)$ through the DNN $u_\pi(\mathcal{s}, a; \omega)$ where ω is the neural network's weight, the DQN algorithm provides an ensemble of RL with neural networks [36]. In the DQN algorithm, the environment is where the agent interacts with particular challenges, while the agent is in charge of learning. The primary goals of the DQN algorithm are to help the agent learn the optimum course of action and maximize the rewards that follow. The agent's job is to furnish selection of an action to take and training the neural network. The environment's job is to finish updating the state $\mathcal{s}_t$ and computing the reward r_t. In order to determine the real value of Q and the estimated value of Q, respectively, the DQN algorithm will produce two multilayer perceptron neural networks with the same structure, known as the evaluation network and target network.

These two neural networks are used by the agent to decide what to do next. For training the evaluation network, a random sample of the agent's experience over the tuples $\langle\mathcal{S}, A, R, P, \Delta\rangle$ is taken from the experience pool. The target-parameters networks are copied from the evaluation network

every iteration (one iteration includes action selection, reward calculation, network structure update, and evaluation network update), and the evaluation network's parameters are continuously updated based on the loss function, ensuring the convergence of the DQN algorithm [36].

The DQN algorithm calculates Q network label u_max as follows,

$$u_{max} = \begin{cases} \theta_t & terminate \\ \theta_t + \gamma \max\limits_{a_{t+1}} u_{targ}\left(s_{t+1}, a_{t+1}\right) & otherwise, \end{cases} \tag{10}$$

Now, we can state the update function for the value function as,

$$u_{eval}\left(s_t, a_t\right) \leftarrow u_{eval}\left(s_t, a_t\right) + \sigma\left[u_{max} - u_{eval}\left(s_t, a_t\right)\right] \tag{11}$$

The update function in the above equation comprises the evaluation network which comprises of the states and actions of the Q network after t transitions.

The weight updating formula for the evaluation network can be obtained as,

$$\omega_{t+1} \leftarrow \omega_t + \left[u_{max} - u_{eval}\left(s_t, a_t; \omega\right)\right] d\omega u_{eval}\left(s_t, a_t; \omega\right) \tag{12}$$

The loss function for the evaluation network in terms of mean square error function employed during training can be obtained as,

$$L\left(\omega\right) = \mathbb{E}\left[u_{max} - u_{eval}\left(s_t, a_t; \omega\right)\right]^2 \tag{13}$$

In Algorithm 1, we provide the detailed execution steps for training the proposed DQN algorithm over the considered heart disease dataset. We can analyze the time complexity of the algorithm proposed in Algorithm 1 as follows: Steps 4 and 10 consider a constant time for binary value of u_max. For different value of t in Step 11, the worst time complexity may be $\mathcal{O}(|t|)$. Therefore, the worst case time-complexity for executing the proposed algorithm is $\mathcal{O}(|t|)$.

Algorithm 1 Proposed DQN-Based Learning

1: **Initialize:** Q network parameters
2: **Set:** Target network u_{targ}
3: **Set:** Agent a
4: **While** u_{max} not converged **do**
5: Set new e-greedy action
6: Select a from s based on policy π
7: **Compute** updated functions
8: $u_{eval}\left(s_t, a_t\right)$
9: ω_{t+1}
10: **If** u_{max} converged **then**
11: **For** t **do**
12: $u_{max} = \theta_t + \gamma \max\limits_{a_{t+1}} u_{targ}\left(s_{t+1}, a_{t+1}\right)$
13: **End For**
14: **End If**
15: **Compute:**
16: $L\left(\omega\right) = \mathbb{E}\left[u_{max} - u_{eval}\left(s_t, a_t; \omega\right)\right]^2$
17: **End While**
18: **Return:** u_{max}
19: **Exit**

D. DELAY MODEL

It is essential to determine the delay of the considered system as most of the tasks pertaining to a healthcare system are time critical in nature and demand on time dissemination of the requested services. Here, we consider the system for switching between the local computing system and mist computing node as a binary variable $\tau_i \forall i \in I$ which can be represented as,

$$\tau_i = \begin{cases} 0, & compute\ locally \\ 1, & offload\ to\ mist\ node \end{cases} \tag{14}$$

The time taken to execute a given task locally can be obtained as,

$$\tau_i^{loc} = \frac{p_{tasks}}{f_i} \tag{15}$$

where f_i denotes the CPU's computing frequency for IoT devices $i \in I$. On the other hand if the task is being offloaded to the mist node, the time taken for its execution can be stated as the ratio between the sum of transmission time τ_i^{Tx}, propagation delay τ_i^{prop}, queuing delay τ_i^{q}, and task execution time τ_i^{exe} to the computing frequency of the mist nodes f_M such that,

$$\tau_i^{mist} = \frac{\sum_{i \in I}\left(\tau_i^{Tx} + \tau_i^{prop} + \tau_i^{q} + \tau_i^{exe}\right)}{f_M} \tag{16}$$

E. ENERGY MODEL

The tasks offloaded by IoT device layer to the mist nodes consumes a substantial amount of energy and depends of several crucial factors like the type of task being executed, and frequency of CPU. Following [41], we can compute the energy consumption by the local computing IoT devices to offload the captured IoT data can be given as,

$$\mathcal{E}_i^{loc} = \rho(f_i)^2 p_{tasks} \tag{17}$$

From the above Eq. (17), the power coefficient pertaining to the architecture of the chip is denoted as ρ.

Further, when n_{tasks} number of tasks are offloaded, the energy consumed by the device can be given as the energy consumed depending on the size of the task to be transmitted s_{tasks},

$$\mathcal{E}_i^{loc} = \rho(f_i)^2 s_{tasks} \tag{18}$$

Hence, we finally obtain the energy consumed to offload tasks n_{tasks} as,

$$\mathcal{E}_i^{Tx} = \mathcal{E}_i^{loc} \times \tau_i^{Tx} \tag{19}$$

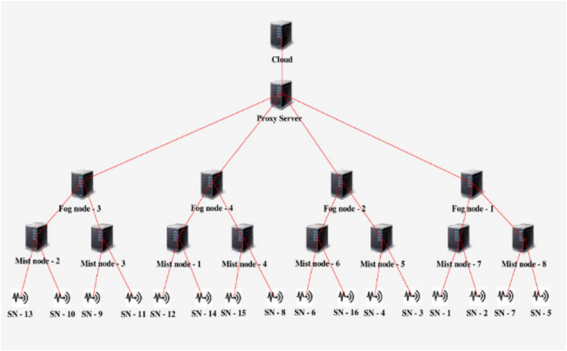

FIGURE 3. Generated topology for proposed mist computing-based paradigm using iFogSim.

V. PERFORMANCE EVALUATION

A. SIMULATION SETUP

We have spoken about the mapping between the elements of various layers in the mathematical model. We incorporate the same in the simulation using iFogSim [33], [34], [35]. We implemented iFogSim using the Netbeans IDE. Classes for our simulation are constructed in iFogSim using the Java programming language. The proposed paradigm's topology is developed. It displays the many application modules that have been created and can be incorporated to execute on various physical setups. Figure 3 displays the constructed topology for Mist-Fog-Cloud layers along with the IoT devices.

The sensor nodes are positioned in layer 1, as can be seen in Figure 3. The sensor nodes gather information about each object's status in the environment and provide it to the mist computing nodes at layer 2. Sensors can be single-dimensional, multi-dimensional, homogeneous, or heterogeneous. The mist layer continuously collects data from the sensors and processes them in a low-cost energy efficient manner. The layer 3 fog nodes receives the data gathered from the mist layer. The grouping of several mist nodes and resource virtualization are managed by the fog organiser. It serves as a distributed link between the cloud and the mist layer. It assigns the resources after receiving the request from an application manager or user, and is also in charge of keeping track of and sharing the available resources. The fog layer handles the processing of sensor and healthcare data. Our experimental setup comprises of four fog nodes connected to the gateway node at layer 4. The simulation environment has different sensor nodes to collect healthcare data heterogeneously networked with the mist nodes. Further the mist and fog computing nodes portray a heterogeneous configuration. The cloud is located in layer 5. Multiple layer 1 sensor nodes are mapped into a single layer 2 mist node, in a many-to-one mapping as shown in Figure 3.

We also see that a single layer 3 fog node maps several layer 2 mist nodes and implements a many-to-one mapping. Finally, the cloud layer in layer 5, which has several cloud instances, is coupled to four fog nodes. The mapping is hence many-to-many.

TABLE 2. Summary of simulation parameters considered for experimental setup.

Parameters	Value
No. of SNs	16
No. of Mist nodes	8
No. of fog nodes	4
CPU frequency of SN	16-84 MHz
CPU frequency of Mist nodes	2.4 GHz
CPU frequency of Fog nodes	4.2 GHz
Transmit power of SN	60 mW
Battery capacity of SNs	1000 J
Average task size (in KB)	450

B. RESULTS

The mist computing model outlined in Section IV is responsible for processing data pertaining to heart patients and gives back prediction results by employing the intelligent DQN algorithm that aid in determining the prevalence of heart diseases, along with the veracity of the claim. Following the work in reference [21], the present study uses the Cleveland heart disease dataset comprising of 14 attributes and observations from 303 records. The 14 attributes account for 13 features and a target attribute which suggests prevalence of heart diseases in the patients' records i.e., 1 denoting heart disease and 0 representing no heart disease. The 13 features entail data carried out through non-invasive clinical trials along with some patient information like age and gender. The target attribute comprises of results from invasive coronary angiogram accounting to 0 and 1 for determining absence or presence of heart disease respectively. The data was acquired by Robert Detrano, M.D., Ph.D. of Cleveland Clinic Foundation.

In this section, different tests are carried out to assess the effectiveness of the suggested DQN algorithm. Our first experiment is a comparative one for heart disease detection. For network training and testing, we exploit the above discussed dataset for heart disease detection. We categorise the considered dataset into instances for prevalence of heart disease and non-prevalence of heart disease in patients of varying age group and gender. There are 4 categories of chest pain i.e., typical angina, atypical angina, non-anginal pain, and asymptomatic patients. Three categories of rest ecg results were presented denoted as 0, 1, and 2 representing normal, ST-T wave abnormality, and probable or definite left ventricular hypertrophy respectively. The slope for peak exercise ST segment was branched into three categories carrying values 1, 2, and 3 representing upsloping, flat, and downsloping graph of the ST segment. Further, the target

TABLE 3. Comparison of performance metrics for QRL,DRL, and DQN algorithms.

	MODELS		
	QRL	DRL	DQN
Precision	0.9112	0.9514	**0.9701**
Recall	0.8914	0.9381	**0.9512**
F-measure	0.9081	0.9498	**0.9761**

variable representing the prevalence of heart disease was denoted as 0 for <50% narrowing of any major blood vessel and 1 for >50% narrowing.

We compare our method to the existing methodologies (viz., QRL AND DRL) using the performance metrics like precision, recall, f-measure, and prediction accuracy for determining the prevalence of heart disease. The precision, recall, f-measure and accuracy can be mathematically represented as [40], [41], [42], [43], [44], [45], and [46],

$$Pr = \frac{T_p}{T_p + F_p} \tag{20}$$

$$R = \frac{T_p}{T_p + F_n} \tag{21}$$

$$f-measure = \frac{2 \times Pr \times R}{Pr + R} \tag{22}$$

$$A = \frac{T_p + T_n}{(T_p + T_n) + (F_p + F_n)} \times 100 \tag{23}$$

where T_p, T_n, F_p, F_n denote the true positive, true negative, false positive, and false negative respectively. The notations Pr, R, $f - measure$, and A denote the precision, recall, f-measure and accuracy of the model respectively.

In Figure 4, we provide the comparative analysis for precision, recall, and f-measure for the considered heart disease dataset. We compare the DQN approach with two well-known existing methods QRL and DRL. It was observed from the experimental outcomes that the proposed DQN approach outperformed other approaches by providing a precision value of 0.9701 as compared to QRL and DRL which incurred precision values of 0.9112 and 0.9514 respectively. The recall for the proposed approach was the highest with value 0.9512 as compared to the other two benchmark approaches which provided recall values of 0.8914 and 0.9381 for QRL and DRL algorithms respectively. Further, the f-measure for the proposed scheme was 0.9761, which outperformed that of QRL and DRL providing f-measure values of 0.9081 and 0.9498 respectively. Table 3 provides the detailed comparison for QRL, DRL, and DQN algorithms corresponding to performance metrics viz., precision, recall, and f-measure.

We present the outcomes for training and testing accuracy over the considered heart disease dataset for varying number of iterations in Figure 5. The experimentation was performed

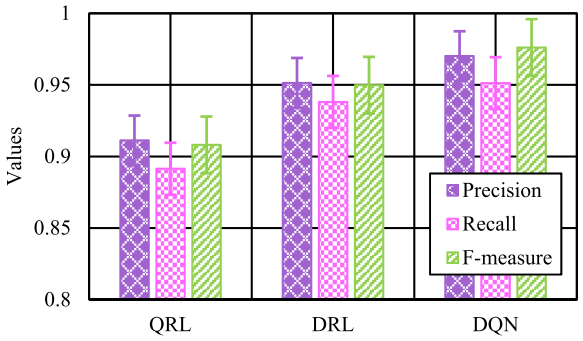

FIGURE 4. Comparison over performance metrics like precision, recall, and f-measure for QRL, DRL, and proposed DQN algorithm.

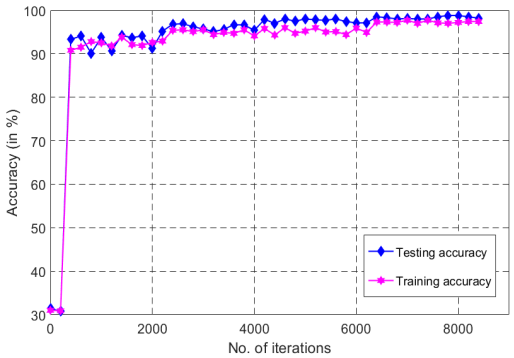

FIGURE 5. The testing and training accuracy for the proposed DQN algorithm over the considered dataset.

over different number of iterations varying from 0 to 9000 to assess the efficacy of the model. It was observed that the model provided fast convergence and the highest training accuracy of 95.6112 % was achieved along with a testing accuracy of 97.6714 %.

Figure 6 provides the comparison of training loss and testing loss incurred for both phases pertaining to different iterations varying within the range 0 to 9000. We observed that the proposed approach provides a loss value of 0.3841 for the testing phase and loss value of 0.4175 for the training phase.

In Figure 7, the comparison for training accuracy was presented over the heart disease dataset with different number of mist nodes for the QRL, DRL and DQN approach. It was observed that as the number of mist computing nodes increases, the training accuracy improves steadily with the addition of each consecutive node. This increase is incurred due to the fact that each node learns over the considered models corresponding to the heart disease data it receives for training and with the increase in number of mist nodes in the network, each node learns based upon the experience of its preceding nodes. As a result, the training accuracy of the models grow for various iterations of the training set because the models tend to over-fit the sample training set. It is further noteworthy to mention that the proposed DQN

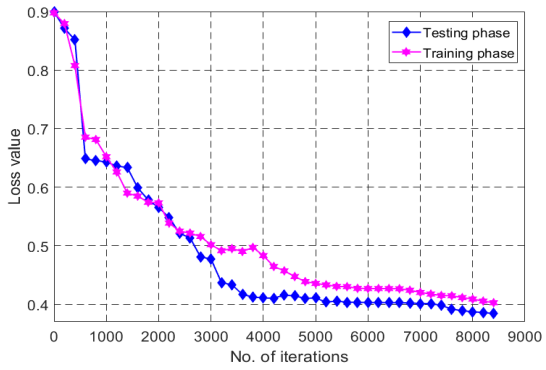

FIGURE 6. The comparison for loss values pertaining to training and testing phases of proposed DQN algorithm.

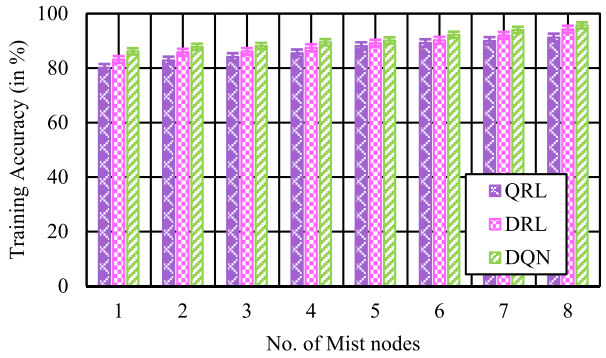

FIGURE 7. Comparison of training accuracy over varying number of mist nodes.

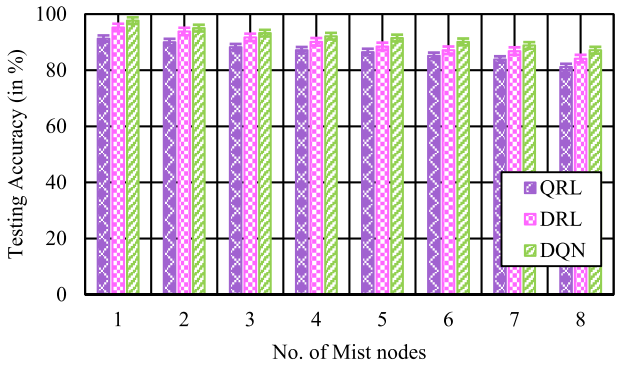

FIGURE 8. Comparative analysis for testing accuracy corresponding to varying number of mist nodes.

approach outperforms the QRL and DRL approaches for all the computing instances of the mist layer.

Figure 8 depicts how testing accuracy varies with the increase in number of mist computing nodes. It is pertinent to note at this point that since each node receives a smaller fraction of training dataset and therefore, cannot approximate to the DQN model, the accuracy of the test declined as the number of nodes increased. Further, the proposed DQN approach's adaptation results in consistently higher performance as compared to QRL and DRL.

TABLE 4. Comparison of energy consumption (in mJ) for QRL, DRL, and DQN algorithms over varying tasks.

No. of Tasks	QRL	DRL	DQN
50	18.4511	17.1347	16.2101
100	21.1261	20.8211	19.6014
150	23.1142	22.4142	20.1972
200	31.4712	30.8712	28.4312
250	35.1214	33.1011	**32.1002**

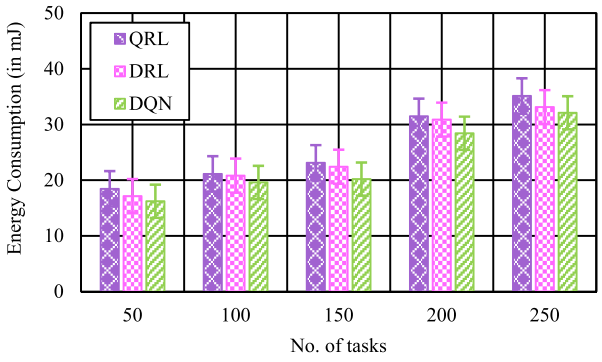

FIGURE 9. The energy consumption (in mJ) for varying frequency of tasks.

We analyse the energy consumption by the mist nodes for building the QRL, DRL, and DQN approach through our simulations. From Figure 9, the performance for the proposed DQN algorithm in convergence with the benchmark algorithms can be observed. From the figure it can be observed that the energy consumption entails an almost linear growth. The performance also depends upon the learning algorithm adopted. However, the proposed DQN algorithm outperforms all the other approaches in terms of minimizing the energy consumption in building the model over a mist computing environment. The highest energy consumed by the DQN algorithm for processing 250 tasks over the mist nodes was observed to be 32.1002 mJ, whereas that of QRL and DRL algorithms were at 35.1214 mJ and 33.1011 mJ respectively. In Table 4, the comparative study for energy consumed by QRL, DRL, and DQN algorithms for processing varying number of tasks over the mist nodes is provided.

We also analyze the delay incurred by each mist node to build the training model by leveraging the QRL, DRL, and DQN approaches for varying number of tasks. Figure 10 shows the performance of the proposed DQN scheme in conjunction with the benchmark approaches i.e., QRL and DRL. From the figure it can be observed that the delay incurred for each algorithm (in milliseconds) witness an increase with the increase in number of incoming tasks. However, the proposed DQN algorithm accounts for the lowest delay of 2.8002 ms in building the model by outperforming the QRL and DRL approaches, which produced a delay of 4.1211 ms and 3.8721 ms respectively. Table 5 provides the comparative analysis between the delay incurred by QRL, DRL, and DQN

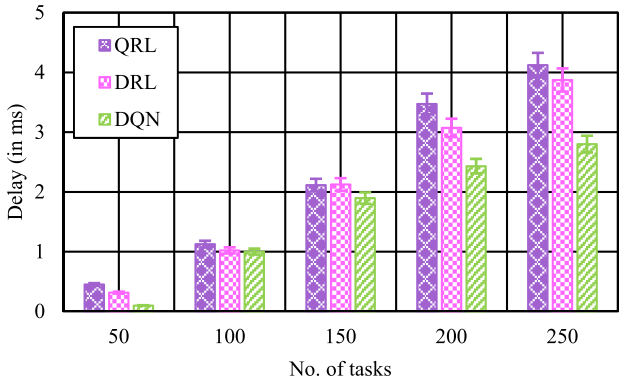

FIGURE 10. Comparison of delay (in ms) for different tasks.

TABLE 5. Comparison of delay (in ms) for QRL, DRL, and DQN algorithms over varying tasks.

No. of Tasks	QRL	DRL	DQN
50	0.4511	0.3147	0.0987
100	1.1261	1.0211	1.0014
150	2.1142	2.1242	1.8972
200	3.4712	3.0712	2.4312
250	4.1211	3.8721	**2.8002**

algorithms for processing tasks of varying length over the mist computing nodes.

VI. CONCLUSION AND FUTURE WORK

In order to determine the prevalence of heart disease for smart healthcare systems, this study proposed a DeepMist framework. The proposed framework exploited the DQN algorithm to facilitate prediction and was built over a low latency energy efficient Mist computing platform. The proposed architecture comprised of four layers namely, the IoT device/application layer, Mist layer, Fog Layer, and Cloud layer. The proposed DQN algorithm provides fast convergence with a prediction accuracy of 97.6714 % and loss value of 0.3841 and outperforms the benchmark schemes considered in this study viz., QRL and DRL in terms of precision, recall, f-measure, accuracy, energy consumption, and delay. Thus it is evident from the simulations and predictive analytics performed in this study that the proposed DeepMist algorithm addresses most of the QoS requirements of present healthcare systems.

In future, we would give more emphasis on determining the optimal CPU utilization of Mist computing layer to determine the task offloading frequency of mist layer to the Fog layer. Further, we would focus on developing task offloading strategies for secure offloading of healthcare data in the Mist-Fog computing infrastructure.

REFERENCES

[1] M. A. Jabbar, S. K. Shandilya, A. Kumar, and S. Shandilya, "Applications of cognitive Internet of Medical Things in modern healthcare," *Comput. Electr. Eng.*, vol. 102, Sep. 2022, Art. no. 108276.

[2] F. M. Talaat, "Effective prediction and resource allocation method (EPRAM) in fog computing environment for smart healthcare system," *Multimedia Tools Appl.*, vol. 81, no. 6, pp. 8235–8258, Mar. 2022.

[3] S. M. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, and K.-S. Kwak, "The Internet of Things for health care: A comprehensive survey," *IEEE Access*, vol. 3, pp. 678–708, 2015.

[4] A. M. Rahmani, "Exploiting smart e-health gateways at the edge of healthcare Internet-of-Things: A fog computing approach," *Future Gener. Comput. Syst.*, vol. 78, pp. 641–658, Jan. 2018.

[5] A. M. Rahmani, T. N. Gia, B. Negash, A. Anzanpour, I. Azimi, M. Jiang, and P. Liljeberg, "Exploiting smart e-health gateways at the edge of healthcare Internet-of-Things: A fog computing approach," *Future Gener. Comput. Syst.*, vol. 78, pp. 641–658, Jan. 2018.

[6] *European Research Cluster on the Internet of Things, IoT Semantic Interoperability: Research Challenges, Best Practices, Solutions and Next Steps, IERC AC4 Manifesto 'Present and Future*, Eur. Commission IERC (Int. Eur. Res. Cluster), 2014.

[7] B. Xu, L. D. Xu, H. Cai, C. Xie, J. Hu, and F. Bu, "Ubiquitous data accessing method in IoT-based information system for emergency medical services," *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1578–1586, May 2014.

[8] L. Jiang, L. D. Xu, H. Cai, Z. Jiang, F. Bu, and B. Xu, "An IoT-oriented data storage framework in cloud computing platform," *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1443–1451, May 2014.

[9] O. Faust, Y. Hagiwara, T. J. Hong, O. S. Lih, and U. R. Acharya, "Deep learning for healthcare applications based on physiological signals: A review," *Comput. Methods Programs Biomed.*, vol. 161, pp. 1–13, Jul. 2018.

[10] S. Tuli, N. Basumatary, and R. Buyya, "EdgeLens: Deep learning based object detection in integrated IoT, fog and cloud computing environments," in *Proc. 4th Int. Conf. Inf. Syst. Comput. Netw. (ISCON)*, Nov. 2019, pp. 496–502.

[11] T. G. Dietterich, "Ensemble methods in machine learning," in *Proc. Int. Workshop Multiple Classifier Syst.* Berlin, Germany: Springer, 2000, pp. 1–15.

[12] S. Tuli, R. Mahmud, S. Tuli, and R. Buyya, "FogBus: A blockchain-based lightweight framework for edge and fog computing," *J. Syst. Softw.*, vol. 154, pp. 22–36, Aug. 2019.

[13] Y. Wang, S. Nazir, and M. Shafiq, "An overview on analyzing deep learning and transfer learning approaches for health monitoring," *Comput. Math. Methods Med.*, vol. 2021, pp. 1–10, Mar. 2021.

[14] S. Ali and M. Ghazal, "Real-time heart attack mobile detection service (RHAMDS): An IoT use case for software defined networks," in *Proc. IEEE 30th Can. Conf. Electr. Comput. Eng. (CCECE)*, Apr. 2017, pp. 1–6.

[15] M. Rajasekaran, A. Yassine, M. S. Hossain, M. F. Alhamid, and M. Guizani, "Autonomous monitoring in healthcare environment: Reward-based energy charging mechanism for IoMT wireless sensing nodes," *Future Gener. Comput. Syst.*, vol. 98, pp. 565–576, Sep. 2019.

[16] N. Constant, D. Borthakur, M. Abtahi, H. Dubey, and K. Mankodiya, "Fog-assisted wIoT: A smart fog gateway for end-to-end analytics in wearable Internet of Things," 2017, *arXiv:1701.08680*.

[17] A. Rajkomar, "Scalable and accurate deep learning with electronic health records," *NPJ Digit. Med.*, vol. 1, no. 1, pp. 1–10, 2018.

[18] S. R. Moosavi, "End-to-end security scheme for mobility enabled healthcare Internet of Things," *Future Gener. Comput. Syst.*, vol. 64, pp. 108–124, Mar. 2016.

[19] I. Azimi, J. Takalo-Mattila, A. Anzanpour, A. M. Rahmani, J.-P. Soininen, and P. Liljeberg, "Empowering healthcare IoT systems with hierarchical edge-based deep learning," in *Proc. IEEE/ACM Int. Conf. Connected Health, Appl., Syst. Eng. Technol.*, Sep. 2018, pp. 63–68.

[20] M. G. R. Alam, M. S. Munir, M. Z. Uddin, M. S. Alam, T. N. Dang, and C. S. Hong, "Edge-of-things computing framework for cost-effective provisioning of healthcare data," *J. Parallel Distrib. Comput.*, vol. 123, pp. 54–60, Jan. 2019.

[21] P. Verma, R. Tiwari, W.-C. Hong, S. Upadhyay, and Y.-H. Yeh, "FETCH: A deep learning-based fog computing and IoT integrated environment for healthcare monitoring and diagnosis," *IEEE Access*, vol. 10, pp. 12548–12563, 2022.

[22] R. M. Abdelmoneem, A. Benslimane, E. Shaaban, S. Abdelhamid, and S. Ghoneim, "A cloud-fog based architecture for IoT applications dedicated to healthcare," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–6.

[23] J. S. Preden, K. Tammemäe, A. Jantsch, M. Leier, A. Riid, and E. Calis, "The benefits of self-awareness and attention in fog and mist computing," *Computer*, vol. 48, no. 7, pp. 37–45, Jul. 2015.
[24] E. M. Dogo, A. F. Salami, C. O. Aigbavboa, and T. Nkonyana, "Taking cloud computing to the extreme edge: A review of mist computing for smart cities and industry 4.0 in Africa," in *Edge Computing: From Hype to Reality*. Springer, 2019, pp. 107–132.
[25] S. Ketu and P. K. Mishra, "Cloud, fog and mist computing in IoT: An indication of emerging opportunities," *IETE Tech. Rev.*, vol. 39, no. 3, pp. 713–724, May 2022.
[26] E. Fazel, H. E. Najafabadi, M. Rezaei, and H. Leung, "Unlocking the power of mist computing through clustering techniques in IoT networks," *Internet Things*, vol. 22, Jul. 2023, Art. no. 100710.
[27] V. K. Quy, N. van Hau, D. van Anh, and L. A. Ngoc, "Smart healthcare IoT applications based on fog computing: Architecture, applications and challenges," *Complex Intell. Syst.*, vol. 8, no. 5, pp. 3805–3815, Nov. 2021.
[28] S. Pallewatta, V. Kostakos, and R. Buyya, "QoS-aware placement of microservices-based IoT applications in fog computing environments," *Future Gener. Comput. Syst.*, vol. 131, pp. 121–136, Jun. 2022.
[29] N. Kumari, A. Yadav, and P. K. Jana, "Task offloading in fog computing: A survey of algorithms and optimization techniques," *Comput. Netw.*, vol. 214, Sep. 2022, Art. no. 109137.
[30] F. Firouzi, B. Farahani, and A. Marinšek, "The convergence and interplay of edge, fog, and cloud in the AI-driven Internet of Things (IoT)," *Inf. Syst.*, vol. 107, Jul. 2022, Art. no. 101840.
[31] B. Kar, W. Yahya, Y.-D. Lin, and A. Ali, "Offloading using traditional optimization and machine learning in federated cloud-edge-fog systems: A survey," *IEEE Commun. Surveys Tuts.*, early access, Jan. 25, 2023, doi: 10.1109/COMST.2023.3239579.
[32] R. Mahmud, K. Ramamohanarao, and R. Buyya, "Application management in fog computing environments: A taxonomy, review and future directions," *ACM Comput. Surv.*, vol. 53, no. 4, pp. 1–43, Jul. 2021.
[33] H. Gupta, A. V. Dastjerdi, S. K. Ghosh, and R. Buyya, "iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, edge and fog computing environments," *Softw., Pract. Exper.*, vol. 47, no. 9, pp. 1275–1296, 2017.
[34] R. Mahmud and R. Buyya, "Modelling and simulation of fog and edge computing environments using iFogSim toolkit," in *Fog and Edge Computing: Principles and Paradigms*. Wiley, 2019, pp. 1–35.
[35] R. Mahmud, S. Pallewatta, M. Goudarzi, and R. Buyya, "IFogSim2: An extended iFogSim simulator for mobility, clustering, and microservice management in edge and fog computing environments," *J. Syst. Softw.*, vol. 190, Aug. 2022, Art. no. 111351.
[36] J. Fan, Z. Wang, Y. Xie, and Z. Yang, "A theoretical analysis of deep Q-learning," in *Proc. Learn. Dyn. Control*, Jul. 2020, pp. 486–489.
[37] G. Zhao, S. Tatsumi, and R. Sun, "RTP-Q: A reinforcement learning system with time constraints exploration planning for accelerating the learning rate," *IEICE Trans. Fundam. Electron., Commun. Comput. Sci.*, vol. 82, no. 10, pp. 2266–2273, 1999.
[38] B. Hengst, "Generating hierarchical structure in reinforcement learning from state variables," in *Proc. PRICAI*, Aug. 2000, pp. 533–543.
[39] D. Shen, G. Chen, J. B. Cruz, C. Kwan, and M. Kruger, "An adaptive Markov game model for threat intent inference," in *Proc. IEEE Aerosp. Conf.*, Mar. 2007, pp. 1–13.
[40] S. Bebortta, M. Panda, and S. Panda, "Classification of pathological disorders in children using random forest algorithm," in *Proc. Int. Conf. Emerg. Trends Inf. Technol. Eng. (ic-ETITE)*, Feb. 2020, pp. 1–6.
[41] S. S. Tripathy, M. Rath, N. Tripathy, D. S. Roy, J. S. A. Francis, and S. Bebortta, "An intelligent health care system in fog platform with optimized performance," *Sustainability*, vol. 15, no. 3, p. 1862, Jan. 2023.
[42] S. Bebortta and S. K. Singh, "An intelligent framework towards managing big data in Internet of Healthcare Things," in *Proc. Int. Conf. Comput. Intell. Pattern Recognit.* Singapore: Springer, 2022, pp. 520–530.
[43] A. Sharma, R. S. Tanwar, Y. Singh, A. Sharma, S. Daudra, G. Singal, T. R. Gadekallu, and S. Pancholi, "Heart rate and blood pressure measurement based on photoplethysmogram signal using fast Fourier transform," *Comput. Electr. Eng.*, vol. 101, Jul. 2022, Art. no. 108057.
[44] S. Pandya, T. R. Gadekallu, P. K. Reddy, W. Wang, and M. Alazab, "InfusedHeart: A novel knowledge-infused learning framework for diagnosis of cardiovascular events," *IEEE Trans. Computat. Social Syst.*, early access, Mar. 2, 2022, doi: 10.1109/TCSS.2022.3151643.
[45] X. Yi, J. Wu, G. Li, A. K. Bashir, J. Li, and A. A. AlZubi, "Recurrent semantic learning-driven fast binary vulnerability detection in healthcare cyber physical systems," *IEEE Trans. Netw. Sci. Eng.*, early access, Aug. 18, 2022, doi: 10.1109/TNSE.2022.3199990.
[46] J. Liu, W. Jiang, R. Sun, A. K. Bashir, M. D. Alshehri, Q. Hua, and K. Yu, "Conditional anonymous remote healthcare data sharing over blockchain," *IEEE J. Biomed. Health Informat.*, early access, Jun. 15, 2022, doi: 10.1109/JBHI.2022.3183397.

SUJIT BEBORTTA (Student Member, IEEE) received the B.Sc. degree in computer science from Utkal University, Bhubaneswar, India, in 2016, the master's degree in computer science from the School of Information and Computer Sciences, Ravenshaw University, Cuttack, India, in 2018, and the M.Tech. degree in computer science and engineering from the College of Engineering and Technology, Bhubaneswar, in 2020. He is currently pursuing the Ph.D. degree in computer science with the School of Information and Computer Sciences, Ravenshaw University. His current research interests include resource allocation in interconnected wireless networks, the Internet of Things, ad-hoc networks and sensor networks, machine intelligence, and randomized algorithm design.

SUBHRANSHU SEKHAR TRIPATHY received the B.Tech. degree in computer science and engineering from Biju Patnaik University, Odisha, India, and M.Tech. degree in computer science from the Bharath Institute of Science and Technology Chennai, Tamil Nadu. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, National Institute of Technology Meghalaya, Shillong, India. He is working as an Assistant Professor with the Dhaneswar Rath Institute of Engineering and Management Studies (DRIEMS), Cuttack, Odisha. His current research interests include cloud computing, fog computing, the IoT, and mist computing.

SHAKILA BASHEER is currently an Assistant Professor with the Department of Information Systems, College of Computer and Information Science, Princess Nourah bint Abdulrahman University, Saudi Arabia.

CHIRANJI LAL CHOWDHARY has been an Associate Professor with the School of Information Technology and Engineering, Vellore Institute of Technology, since 2010. His research interests include computer vision and image processing.