SSRG International Journal of Electrical and Electronics Engineering
ISSN: 2348-8379/ https://doi.org/10.14445/23488379/IJEEE-V9I12P114

Volume 9 Issue 12, 156-166, December 2022


*Original Article*

# Latency Reduction in Medical IoT Using Fuzzy Systems by Enabling Optimized Fog Computing

S. Aiswarya[1], Angelina Geetha[2], K. Ramesh[3], S. Sasikumar[4], D. Sheema[5]

*[1,2,4,5]Hindustan Institute of Technology and Science, Chennai, India*
*[3]Sri Krishna College of Engineering and Technology, Coimbatore, India*

*[1]Corresponding Author : aiswarya.sudha@outlook.com*

Received: 24 October 2022 Revised: 29 November 2022 Accepted: 13 December 2022 Published: 25 December 2022

***Abstract*** *- Fog computing technology is an emerging computing method that functions in a distributed decentralized environment. Cloud computing features are being brought closer to edge devices with fog computing. Healthcare IoT devices should benefit from this approach by achieving minimum latency requirements. A variety of devices in healthcare produce a huge amount of data. The massive volume of data generates network congestion and high latency due to high traffic. With IoTs and cloud servers transmitting data at high rates and having big hop counts, round-trip delays can make data useless and insufficient for users. It is essential to have real-time data for medical applications that are time-critical. Traditional computing servers cannot serve IoT devices at the other end of the medical IoT chain because of their latency requirements. So, the data transmission of different latencies like computation, communication, and network must be reduced. IoT must operate at low latency since data transmission must occur in real-time. This requirement can't be achieved through cloud computing. Due to data volume and factors related to Internet connectivity, analyzing and acting on data can result in high network latency. Fog computing makes it possible to store, process, and examine data from cloud computing at the network edge. As mentioned earlier, the current work presents an innovative solution to the problem. With the help of fog computing, the fuzzy-based reinforcement learning algorithm is integrated with an analytical model. The objective is latency reduction for IoTs in healthcare, including cloud servers. The projected smart fog computing model and algorithms combine fuzzy inference with reinforcement learning and other neural network development methods to allocate and select data packets. A simulation approach is tested using iFogSim and Spyder. Simulated results show the proposed Optimized Latency Fog Computing (OLFC) model has 52% and 30% minimal latency compared with the existing iFogStor and FC models.*

***Keywords*** *- Fog computing, Internet of things, Big data, Healthcare, Fuzzy Systems, Cloud computing.*

## 1. Introduction

The healthcare industry has also evolved with the development of industrial technologies. In the present scenario, everything revolves around smart devices and industry 4.0 [1]. Technological advances like Artificial Intelligence, IoT and big data have made smart devices essential for detecting diseases. During today's hectic era, people pay less attention to their health due to their daily busy lives—health declines due to failure to maintain routine medical check-ups. An unhealthy individual will have a lower level of productivity. The well-being of a country's people directly affects its overall growth and should not be neglected in the drive for development. Providing people with advanced healthcare facilities is of utmost importance to improving their lifestyles. The evolution of numerous technologies has improved the quality of life for human beings, which may allow persons to get a precise diagnosis at a low cost and within a short timeframe through outsourced services on the web. A lot has changed in healthcare along the way to alleviate stress within the medical institution and organize healthcare environments. There have been numerous challenges in the entire transformation of healthcare, such as preserving health records, ensuring global communication, and providing a faster problem-solving process. As a result of challenges, the management of huge amounts of data has emerged as a crucial issue. Healthcare 4.0 has integrated advanced technologies to increase accessibility, speed response times to access data, and safeguard sensitive information.

Fog computing involves the latest technologies and acts as an intermediatory between all the different technologies to manage diverse aspects of healthcare data. In healthcare, many IoT devices are being used, resulting in a tremendous amount of data. Cloud servers are used globally to analyze, store, and pre-process data due to their large volume, diversity, and integrity. Currently, communication along IoTs in healthcare can only be accomplished through the cloud [2]. With cloud computing, healthcare IoT devices are handled efficiently by removing computational tasks that drain batteries [3]. IoT healthcare devices generate data that can be analyzed, filtered, pre-processed, and accumulated only in the cloud. Cloud-based IoT devices, however, have certain limitations. The growing transmission and deterministic nature of these enormous sizes of data have also increased

**Table 1. Requirement of E-Health QoS**

| E-Health services | Health applications | Media type | Delay (max) |
|---|---|---|---|
| Audio communication(real-time) | Audio conferencing (The end users and physicians) | Audio | < 150 ms (single way) |
| Video communication (real-time) | Video conferencing (The end user and physicians) | Video | < 250 ms (single way) |
| Robotic services (real-time) | Telesurgery (remote-based) | Data from robots, audio, and video | < 300 ms (round-trip delay) |
| Patient monitoring (real-time) | Vital signs and health data transfer of patient | Sensor data (Biomedical data) | Apps < 300 ms (real-time ECG) |

cloud computing reaction times. As a result, end users suffer higher service latency reaction times. In addition, the likelihood of an error occurs greatly when data is transmitted over a network in large quantities. Latency and packet loss are proportional to data transmitted by IoT devices to cloud servers [4]. As a result, end users experience poor QoS. Therefore, it is unnecessary to utilize cloud-scale processing and storage in time-critical IoT applications. A greater focus should be placed on extreme time-bound selections regarding IoT devices.

Time-critical healthcare IoT applications require minimum latency with good quality of service, which was one of the primary motivations for the study. It is simply not feasible to fulfil all these requirements in the cloud. The physiological states of patients change over time, so it is necessary to make rapid decisions and respond quickly to remote patients. Unusually unpredictable network conditions can increase latency. Since the latency is so high, it is impossible to return the patient health data in real-time [5][32]. Thus, they are invalid, inaccurate, and unreliable. A more significant problem arises when cascading-based data is processed—for example, the processing of electrocardiogram (ECG) and electroencephalogram (EEG) signals. The response time of services from medical IoT varies from milliseconds to microseconds. The quality of requirement for the medical data is as follows—the quality requirement of medical data transfer rate for Audio, video, and ECG as follows. Latency should be between 150ms-400ms, 150ms-400ms, and 1 second, respectively. The data rate should be 4-25 kbps, 32-384 kbps, and 1-20 kbps, respectively.

Table 1 above shows the data transfer rate quality required for the various medical services.

The earlier studies on fog computing used standard data communication models between cloud services and IoTs; however, in the current scenario, a sophisticated, smart infrastructure is essential to act as a gateway among the Cloud and IoT. A gateway connected to healthcare IoTs and cloud servers reduces the computation, communication, and network latency so that Individual health data can be collected in real-time. A gateway connected to medical IoTs and cloud servers reduces the computation, communication, and network latency for real-time individual health data collection.

The rest of the paper is ordered as follows; related works explains in section II. The proposed IoT-based system model for the patient health data classification is outlined in section III, followed by the algorithms and flowchart—the results and detailed discussion in section IV, along with the simulation analysis results. Finally, section V concludes the paper with future direction discussions.

## 2. Related Work

This segment examines the effects of high latency, network usage, and energy consumption on IoTs, cloud and Fog computing in depth, compares the existing works and provides insight into the new works.

Author Li et al. [7] showed a method for providing IoT and FC services based on service popularity and smart resource partitioning. They addressed the issue of resource efficiency and computing on fog nodes. This work uses IoT and fog servers to decrease delay, response times and fault tolerance. Also, Alam et al. [8] presented a basic block-offloading methodology for deploying mobile codes on decentralized fog networks with geographically distributed users. The blocks were migrated using the RL technique in a distributed multi-agent environment. High latency and processing time were reduced as a result. Kao et al. [9] described using a new technique called Hermes for time-critical issues in mobile computing. Optimizing task assignment for limited resource devices is the purpose of this technique. The offloading of computational tasks was the basis of this technique. In their work, Nishala et al. [10] described a technique called Hipster designed to satisfy end-user requirements in terms of QoS. Both heuristics and RL methods were used in the technique. Cloud computing latency has been reduced for time-sensitive workloads using machine learning effectively. Unfortunately, the author's study did not mention cloud servers' high communication latency. A technique called iFogStor was proposed by Naas et al. [11] to solve the problem of high latency with time-critical IoT applications. This methodology incorporated FC principles. For iFogStor, the problem of data placement is a generalized assignment problem (GAP). The authors recommended a heuristic and accurate integer programming approach to solve the problem. This approach will require more precise models and planning for time-sensitive IoT applications. In addition to the other edge-cloud-FC computing, Pan et al. [12] discussed developing and current IoT application technologies. The authors pointed out several problems, such as high latency and data traffic with IoT. In addition to their survey analysis, they conducted a core study on emerging and current technologies. While this research addresses latency minimization, it does not deal with practical implementations. Cao et al. [13] proposed reducing mobile device energy consumption and bandwidth usage using machine learning algorithms. In the cloudlet environment, they discussed offloading computational tasks between multiple users. The fog concept was explored in Brogi et al.'s [14] proposal to diploy QoS infrastructure in the fog environment for IoT. This proposal discussed various challenges, including data distribution, scalability, adaptive placement, and segmentation in the IoT-Cloud infrastructure. Nevertheless, both works [13] and [14] did not address the problems associated with the high computing and network latency between the IoT and the cloud. As Mahmud et al. [15] pointed out, healthcare applications suffer from high latency and big data transmissions. Their proposal includes a cloud-based fog service and an architecture for medical care applications. In order to optimize data communication and reduce latency, as well as to reduce power consumption, the obtained results were analyzed. The experiments result in improvements in cost efficiency, network delay, and energy consumption. The hybrid bio-inspired algorithm proposed by Rafique et al. [16] minimizes reply and execution times in the IoT-fog-cloud environment. The hybrid algorithm merges cat swarm and particle swarm optimization techniques. Resources were altered, and the modified algorithm handled task scheduling in fog nodes. The RL techniques will be used to manage resources in IoT – A fog environment in the future. Amit et al. [17] suggest a novel Intelligent Multimedia Data Segregation (IMDS) method in the fog environment in which multimedia data is segregated from the model for calculating total latency (transmission, computation and network). The simulation results indicate improved e-health quality by improving classification accuracy by 92% and reducing latency by approximately 95% compared to the pre-existing model. The model based on the fog environment presented by Shukla et al. [33] combines fuzzy logic with reinforcement learning. Healthcare IoT architecture with three tiers. The ideal network latency is as low as possible. The work was simulated using the iFogSim simulator. Aiswarya et al. [19] incorporated fuzzy logic and reinforcement learning into a 3-tier architecture to develop an analytical model for medical IoT. The model was tested using the iFogSim simulator and effectively reduced the latency. Based on measuring the improvement potential of VFNs for improvement of SFC scalability, Dinh et al. [20] recommend a cost-efficient availability guaranteed deployment scheme for IoT services over fog core cloud networks. Additionally, several techniques are presented for placing a redundancy layer for VNFs. Compared with the existing methods, the obtained analysis and simulation results indicate significant improvements in cost-efficiently and scalability.

In their novel fog-centric secure cloud storage system, Ahsan et al. [21] provide protection against unauthorized access, modification and destruction of data. Data can be concealed using the new X or-combination technique to prevent illegitimate access. Additionally, to facilitate modification detection with a higher probability, proposed a technique based on hash algorithms. La Quang Duy et al. [34] provide two case studies showing how device-driven and human-driven intelligence can work together to reduce energy consumption and latency in fog computing. First, the MAC layer adapts sensor devices based on low-latency user behaviour detected by machine learning.

**Table 2. Comparative study of Latencies (Computation $L_{cp}$, Communication $L_{cm}$, Network Latency $L_{nw}$)**

| Author | Technique | $L_{cp}$ | $L_{cm}$ | $L_{nw}$ |
|---|---|---|---|---|
| Li et al. [7] | SPSRF | x | ✓ | x |
| Alam et al. [8] | RL with a primary offloading mechanism | ✓ | ✓ | x |
| Kao et.al [9] | Hermes | x | ✓ | x |
| Nishala et.al [10] | Hipster | ✓ | x | ✓ |
| Naas et.al [11] | iFogStor | ✓ | x | ✓ |
| Pan et al. [12] | Edge computing | x | x | x |
| Cao et al. [13] | Computation offloading | x | ✓ | x |
| Brogi et al. [14] | FC model with a Heuristic approach | x | ✓ | x |
| Mahmud et.al [15] | Cloud-fog | x | ✓ | x |
| Rafique et al. [16] | Hybrid bio-inspired algorithm | ✓ | ✓ | x |
| Dinh et al. [20] | FC cost-effective scheme | x | ✓ | x |
| Ashan et al. [21] | Fog centric-cloud model | ✓ | x | x |

The second case study examines task offloading as a method for multi-node EU devices to select their offloading decision based on energy and latency objectives while minimizing battery consumption. We found that fog computing can be tackled with a huge amount of previously untapped intelligence. Vilela et al. [23] presented a comparative study of the cloud, the conventional computing model and the new fog computing concept in the health sector. Studies the fog computing model's role in the medical field, highlighting its main parameters. It shows the development and implementation of an online health monitoring system using fog technology. Development of a new healthcare system based on the performance assessment of the proposed solution.

To compare and analyze the proposed method with different approaches used by other researchers, selected different approaches used as a baseline. Here uses the computation latency ($L_{cp}$), Communication Latency ($L_{cm}$) and Network Latency ($L_{nw}$)

Table 2 shows the comparison among the different technologies projected by the authors in terms of computation, communication and network latencies. IoTs and the cloud can be connected more effectively by implementing the strategies discussed in this article. This will minimize network traffic, latency, and energy consumption. Providing a middleware gateway allows data to be sent between cloud servers, IoT devices, and users. Despite this, most existing works do not implement latency minimization in real-world IoT and cloud servers. The majority of these techniques are based on conventional fog computing (FC) methods. Health IoTs rely heavily on the techniques mentioned above. Our proposed approach will be compared to these existing techniques for the above reasons.

This section identifies the limitations of existing IoT fog-cloud techniques. The latency and bandwidth of medical IoTs are also argued to be high and unfeasible due to high computation, network and communication latency. Healthcare IoTs suffer from high latency, which delays the transmission of Individual Health Data. The cloud computing methods and middleware gateways cannot meet healthcare IoT latency and quality of service requirements. IoT in healthcare has not been extensively studied to reduce round-trip time delays between end to end. The high latency is minimized using Neural network progress strategies in conjunction with an analytical model. The proposed model and algorithm meet the requirements of quality of healthcare IoTs.

## 3. Proposed Work

Optimized Latency Fog Computing (OLFC) proposes an IoT-based health model for patient classification using Health Data Classification (HDC) algorithm. Data selection and allocation with the help of RL and greedy algorithm. Using the Fuzzy Inference System (FIS) classification process, the data transmitted by medical IoT devices are divided into low-risk, normal, and high-risk categories. In fog servers, individual health data are assigned to virtual machines through RL. NNs are used to select time-sensitive data, which is then sent to end-users in a timeframe. Data packets are distributed and allocated among other nodes and end users by fog nodes in virtualization. Nodes at the end of the network are connected to fog nodes through which data can be retrieved. A master fog node controller is used to allocate and distribute data packets according to topology information [24].

Fog networks consist of interconnected nodes that fog masters connect. In this study, we investigate the use of fog computing to allocate data packets to machine learning models using a progressive approach. Nodes can transfer data packets to minimize network traffic and latency. An average response time can be calculated using the CPU-queued data packets, which can be used as a node traffic index. In addition to gathering traffic information and assembling queue positions, fog nodes can make decisions and serve end nodes. The master node creates a network table. Nodes receive requests from the master fog node indicating whether or not to move data from one node to another. This will result in the data being sent to the neighbouring node for selection based on the time and the required data. This study aims to reduce latency and network traffic while selecting time-sensitive data. A 3-tier IoT- based patient classification model based on RL and Q-learning fuzzy inference rules is shown in Figure 1. The model can produce the optimal healthcare solution based on RL and neural network (NN) techniques and real-time notifications.

Computation delays, communication delays, and network delays contribute to total latencies. The proposed health IoT model has three layers: the health IoT, Fog, and the cloud layer. The sensor component of the health IoT layer generates an individual Health Data (IHD) metric. According to their level of risk, the fuzzy inference method [19] is categorized into low-risk, normal, and historical-risk methods. The classified data is stored in the fog computing layer. A time-sensitive set of data was selected using RL. IHD is allocated by virtualizing a fog server. The fog layer is used to transmit IHD instantly to end users. Patients' historical data is stored in the cloud layer for future reference. Fog servers notify end users in real-time when a problem is detected. Fog server and layer virtual machines control data transmission through the gateway.

The data can also be selected and allocated with the help of reinforcement learning and NN techniques. It performs the functions based on the inputs given to the Fuzzy Inference System [35]. Connectors are used in conjunction with IF…. THEN rules. A fuzzy membership function and predefined FIS rules are used here to categorize individual health data. An approach based on fuzzy Q-learning is used to model fog networks. The computation latency between distributed fog servers must be as low as possible to process and upload the entire data packet. With greedy methods, the decision is made when data packets are uploaded, which reduces system latency.

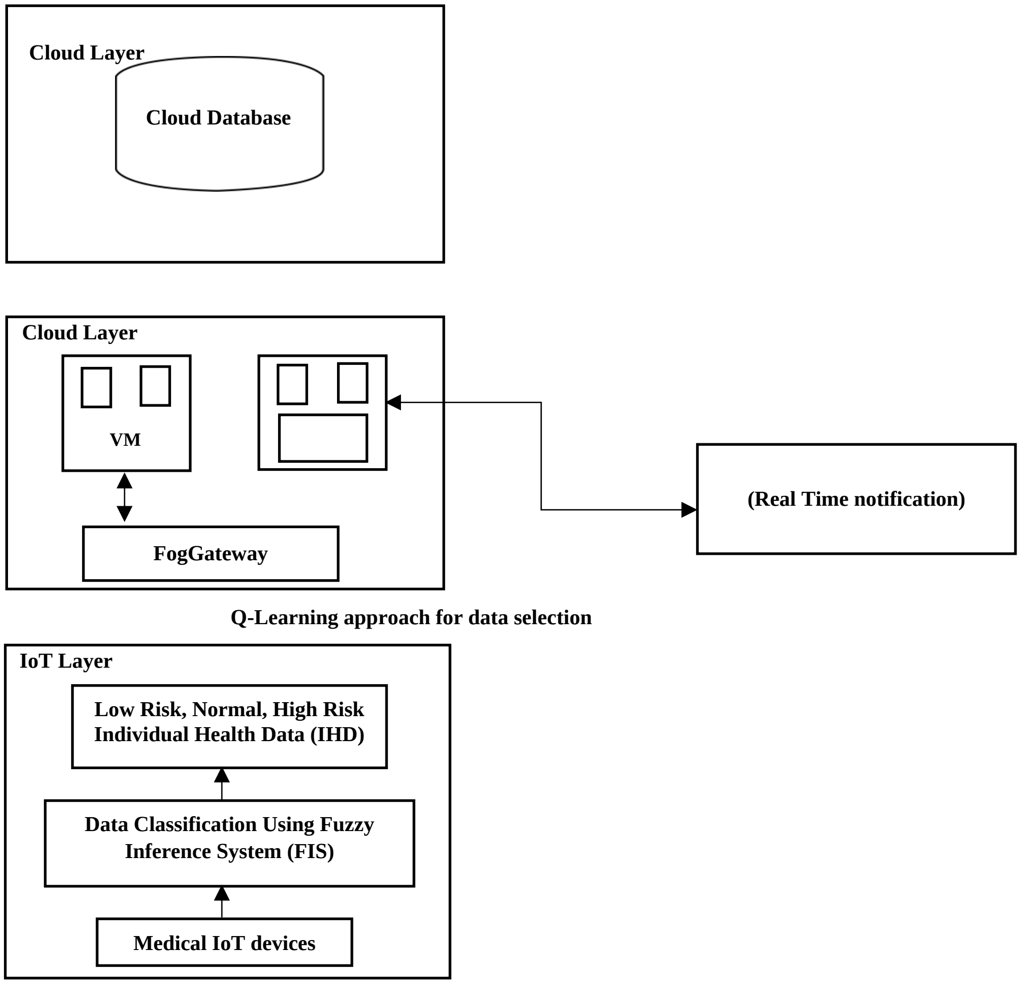


**Fig. 1 IoT-based three-tier patient classification model**

### Algorithm 1: Health Data Classification (HDC) Algorithm

### Patient Health Data Classification using Fuzzy Inference System (FIS)

Step 1: *With inputs and its member function $\mu_a$ the system is created.*

Step 2: *Using the member function, $\mu_a$ (Heart rate1), $\mu_a$ (ECG$_1$), get the health status as $\mu_a$ (normal) or $\mu_a$(Low- risk) or $\mu_a$(high- risk)*

Step 3: ***IF** (Health status = $\mu_a$(high-risk))*

***THEN** send a real-time notification to $f_G$ using SPARK as RTA after getting geo-location*

Step 4: ***ELSE IF** (Health Condition= $\mu_a$(low risk))*

*THEN the $P_{id}$ is sent to $f_G$*

Step 5: *END*

Notations used in this algorithm are, Fog gateway for data packet allocation as ($f_G$), fuzzy inference membership function ($\mu_a$), Real-Time Analyzer (RTA) and Patient ID ($P_{id}$).

This proposed model uses a fog server system as the training environment. An action selection function is present in a model with RL (i.e., selecting the data packets and allocating them to fog servers in real-time), which chooses actions according to the system state. System state can be defined using values. Data packet complexity, size, remaining packets in fog storage, times it takes from the last uploading moment until the present, and the final packet requirements make up these values. The fog server must be uploaded with the data packets to allocate and process the previous packets. Fog server computation latency is measured and reported as a [k x 1] vector. In the next step, determine the time it will allocate if the server deems the arrival packet acceptable. Additional vectors of similar size [k x1] store the value. The combination of the 2 vectors above yields a [2k x 1] vector illustrating the system's state at any given moment. To allocate and send data packets, fog servers have a computational latency of milliseconds. During the allocation process, data packets require a minimum number of megacycles to be transferred between servers, resulting in a deviation in latency. To determine the system state, we calculate the latency for data packet allocation, processing, and transmission in vector form.

When an RL model is utilized, the NN is used to select the actions. NN systems operate based on state inputs. There are 2k nodes in the input layer of the NN, and the hidden layer nodes are interconnected. The input layer consists of 2k nodes, and the state size is [2k x 1]. There are M nodes in the hidden layer, denoted by $H_k$, k== {1,2, …N}. Thus, the hidden and input layers have a [N x k] relationship. Every packet is given a weight, and all the weights are stored within a matrix W(1). The sum of all the weights and inputs determines the value of node $H_k$ in the hidden layer. A hidden layer in the $J^{th}$ node represents a summation of all the inputs and weights. An indication of the training process can be found in the number of nodes in the hidden layer. A hidden layer is attached to the NN output nodes, the softmax layer [26]. The output layer has a capacity of [K x 1]. W(2) represents the total weight of the two layers of the network.

Node values are calculated at the end of the layer. The time-sensitive IHDs are transferred once all the nodes have been calculated. To transfer the time-sensitive IHD, fog server fog (i) is selected, where i = 1, 2 ------K. Based on the chances of each server receiving the data packet, the one with the highest probability will receive it. This latter calculation normalizes. Based on the exponential functions of the input numbers, we can translate the probability distribution for k real numbers into k probabilities. Softmax in RL is used to convert node values into probabilities. NN and RL commonly use it.

To increase the capacity of the fog server for selecting and allocating data packets, we restore the NN as part of upgrading the RL model for successive rewards. Current machine learning algorithms for updating NNs commonly implement backpropagation [27], which is feasible for rewards of either 0 or 1. Since our long-term reward is not known, backpropagation is not applicable. Training neural networks are achieved by neuroevolution (NE), also known as neural network evolution. A new generation is generated from the NN allocated to each iteration. The NN is derived from this generation. A higher reward is used to select children for the NN to be renewed [36]. Evolution strategies are used to update the neural network. This accepted and recognized algorithm can also be used to apply NE.

Algorithm 2 demonstrates real-time allocation and selection of data packets. Data packet allocation is based on minimizing the latency of the schema through the greedy algorithm. Evolution strategies are used to update the NN in an RL environment. A Gaussian noise sum is applied to each weight in the network for each repetition, forming M children of the NN. The RL model by K data packets rewards the NN's children based on means ($Mean\_reward_i$) over k actions. Each of these actions is performed by one child in the NN.

**Algorithm 2: Selection and allocation of data packet using RL with Greedy approach**

Step 1: *i/p learning rate ($\alpha_{in}$), service rate (sr), discount factor ($d_i$), distance vector ($V_t$), exploration policy (Є), arrival rate-data packet (βi), and weight matrix of parent NN (W), No. of children (J) and i = 1, 2*

Step 2: *O/p Data allocation table (Q)and Parent NN with supreme performance.*

Step 3: *Set Q (s, a) = 0(Ɐs Є Si) (Ɐa Є Ai(a)), iteration = 0, and*

*s = (1,1{(Q1 − − − − − QN)} | (Qi = 0)*

Step 4: *While (iteration ≤ max iteration) do*

Step 5: *Apply greedy algorithm on selected a Є Ai(a)*

Step 6: *Assign the data packets corresponding to action a and observe the following state 's' and reward 'r'.*

Step 7: $Q(s,a) \leftarrow (1-\alpha_{in})Q(s,a) + \alpha_{in} [R_i(s,a) + d_i max_{a' \epsilon A_{is'}} Q(s', a')]$

Step 8: $s \leftarrow s'$

Step 9: *iteration ← iteration + 1*

Step 10: *For iteration in a specified range, do*

Step 11: *For J in range N do*

Step 12: *Evaluate*

Step 13: *Child(J) = Parent NN + random noise (N0), [W (J) = W+ noise]*

Step 14: *Compute Mean_reward*

Step 15: *Gain $^{(J)}$ = Reward $^{(J)}$ – Mean_reward (where J= 1, -------, N)*

Step 16: *Evaluate parent NN*

Step 17: *End*

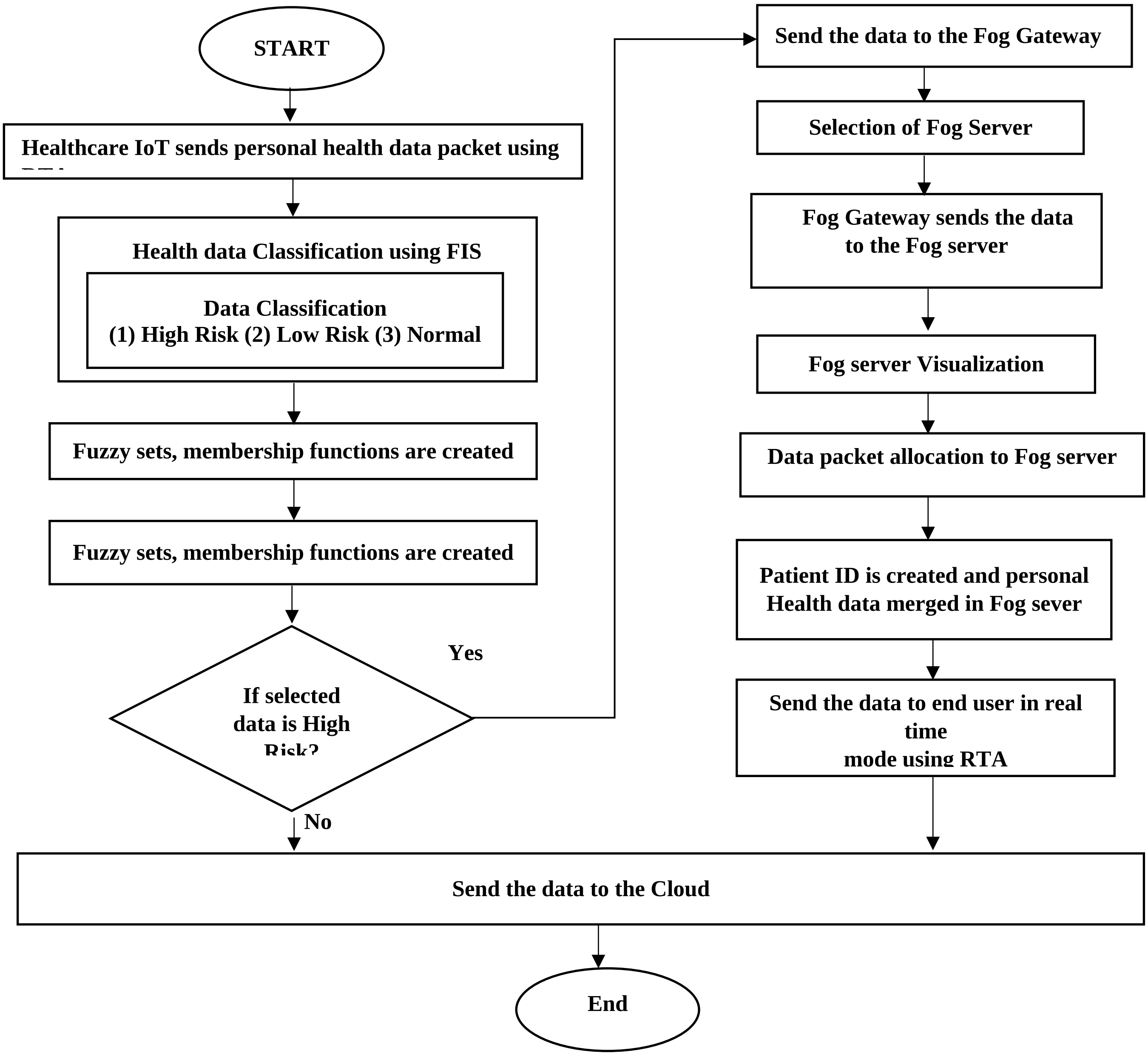


**Fig. 2 Flow chart for the real-time communication of data packets in a fog environment**

A data allocation table Q is generated following the computation of the maximal performing parent NN. Algorithm 2 presents the proposed Q-learning algorithm. It explores the optimum reward field for RL models involving data packet selection and allocation. Figure 2 illustrates how an IoT data packet of healthcare can be communicated in real-time via fog computing the proposed algorithm flow.

Health data is produced from various IoT devices using the real-time analyzer. Based on a range of fuzzy values, fuzzy sets are created using the Fuzzy Inference System. Then, using the fuzzy rule output results and member functions (heart rate and ECG) to identify the state of health as normal, low risk, or high risk. Membership function - $\mu_1$ used to denote the heart rate and ECG.

## 4. Results and Discussion

The proposed model Optimized Latency Fog Computing (OLFC) is evaluated and analyzed with this section's proposed machine learning algorithm. The machine learning algorithm and the fog-based model were validated through numerical simulations. Using a support vector machine (SVM) [29][30], HDCs were subjected to predictive analysis to determine their robustness. Performance measures such as sensitivity, accuracy, Positive Predictive Value (PPV), and Negative Predictive Value (NPV) demonstrate the validity and efficacy of the algorithm.

A simulation was performed to examine the OLFC model based on the projected algorithm. The baseline minimizes Fog environments' latency, network usage, and energy consumption. Here, the tools iFogSim (open-source s/w) and the Spyder (Python-based) editor were used to simulate the FC-based analytical model.

The main parameters analyzed in this study are listed below

- Latency
- Bandwidth
- Energy Consumption

All these parameters are checked with the Load, i.e., the number of user requests from the connected devices.

### *4.1. Latency*

In network communication, latency, also called lag, represents delays in data exchange. Data latency mentions the time it takes to capture, transmit, process, receive, and decode a packet. The time taken for medical data transmission in healthcare is more important than in other sectors. Every minute is a matter of life. Here the latency is calculated in milliseconds. Total latency can be calculated as a summation of computation, communication and network latencies. Computation latency is the total waiting time and service time. Communication latency is calculated by calculating the time required for a data packet to travel between the two nodes. i.e., round trip time among the end user and the fog node. Network latency calculates the delay resulting from the total number of packets sent between sensor networks and fog networks.

Total latency can calculate as follows,

$$L = L_{cp} + L_{cm} + L_{nw} \quad \text{-------------------- (1)}$$

Fig. 3 shows the latency comparison between the iFogStor, Fc, and the new Optimized Latency Fog Computing (OLFC) model. It is found from the results that the OLFC model has 52% minimal latency compared with the existing iFogStor and 30% more than the FC model.

### *4.2. Bandwidth*

The highest amount of data can be transmitted over an internet connection in a given period. A connection's bandwidth is its speed, measured in megabits per second (Mbps). A network can transfer massive data between connections or devices within a specific period. Suppose the higher the bandwidth higher the data transfer rate. The fog environment provides better bandwidth than the cloud. Bandwidth can be calculated as follows:

$$Bandwidth\,(Mbps) = (TD * (100 / DDR) * 8192) / (RWT * 3600) \quad \text{---------------(2)}$$

Where,

TD - Total amount of data (GB)

DDR - Data deduplication ratio

RWT – Replication Window Time length (Hrs)

Fig. 4 shows the Bandwidth comparison between the iFogStor, Fc, and the new OLFC model. It is found from the results that the OLFC model has lower bandwidth utilization by 1.5% and 0.7% compared with the existing methods, iFogstor and FC model, respectively.

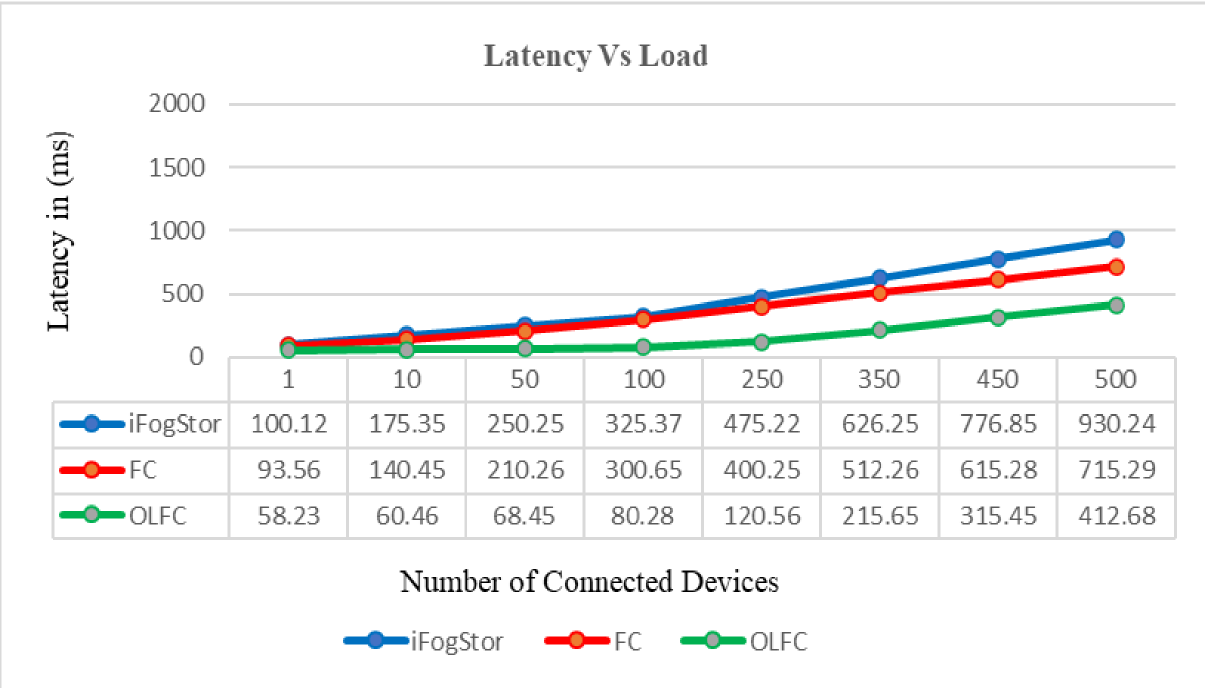


**Fig. 3 Network latency Vs Load**

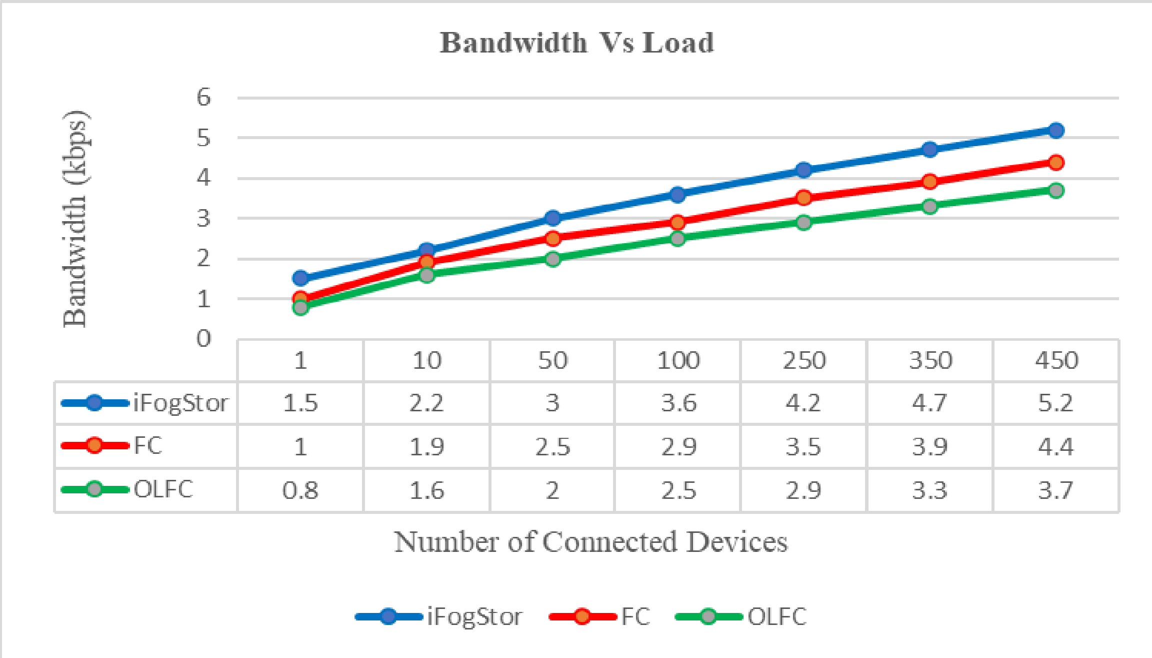


**Fig. 4 Bandwidth Vs Load**

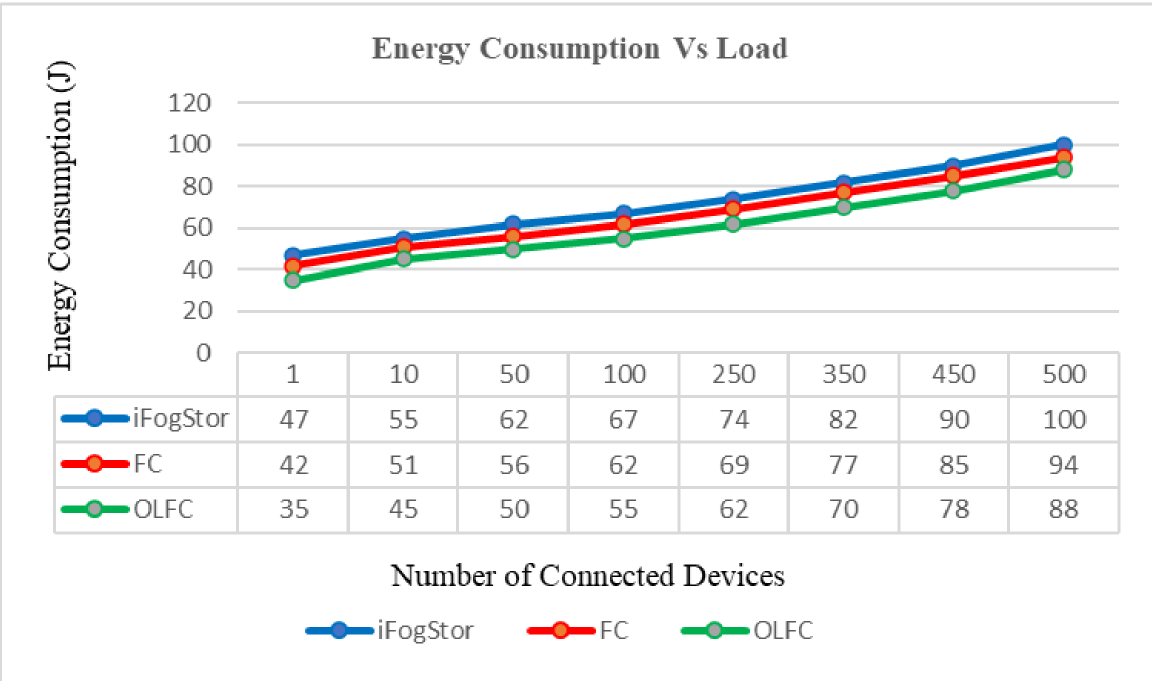


**Fig. 5 Energy consumption Vs Load**

### *4.3. Energy Consumption*

Distributed systems are often constrained by their energy consumption. Medical applications are latency-sensitive, making fog computing one of the best solutions for computation tasks. As a result, it is crucial to analyze medical jobs' impact on the energy consumption of fog resources. In general, energy consumption is influenced by the way in which it is utilized. For fog nodes to process sensor data, they consume an increasing amount of electricity. The cloud computing model consumes more energy than the fog computing model. When the number Load increases, energy consumption also increases.

Fig. 5 shows the energy consumption between the iFogStor, Fc, and the new OLFC model. It is found from the results that the OLFC model has 12% lower energy consumption compared with the existing iFogStor model and 6% lower than the FC model.

## 5. Conclusion

Healthcare IoT devices produce a huge size of data. End users are delayed in receiving services in an IoT-cloud environment. Traditional cloud services do not meet the latency demands of healthcare IoTs. Hence, the presented fog-based OLFC model will minimalize the latency among IoTs, users at the endpoint and cloud servers. A fuzzy-based hybrid RL algorithm follows RL algorithms incorporating NN evolution strategies. The projected algorithm is used in a fog environment to allocate and select healthcare IoT data packets. FIS and linear SVM are used to classify healthcare IoT data. RL and NN evolution strategies assign and select data packets in fog nodes. The following parameters were used to investigate high latency: communication, computation and network latency in milliseconds, bandwidth in Mbps and Energy consumption in joules. Simulating the proposed OLFC algorithm produced better results with 52% and 30% lower latency, 1.5% and 0.7% lower bandwidth utilization, and 12% and 6% lower energy consumption than existing iFogStor and FC methods, respectively. Therefore, the proposed approach is optimal, suggesting that it can be applied to IoT in healthcare. By reducing latency among healthcare IoT devices and cloud servers, the proposed algorithm reduces healthcare IoT costs and improves patients' outcomes. The proposed model can be implemented for detecting the patients' early warning score in a future enhancement. Early medical warning score involving FIS's real-time transfer of patient health data to medical agencies and doctors has not yet been conducted with biomedical data analysis.

## References

[1] Aparna Kumari et al., "Fog Computing for Healthcare 4.0 Environment: Opportunities and Ehallenges," *Computers & Electrical Engineering*, vol. 72, pp. 1-13, 2018. *Crossref*, https://doi.org/10.1016/j.compeleceng.2018.08.015

[2] B. Farahani et.al., "Towards fog-driven IoT eHealth: Promises and Challenges of IoT in Medicine and Healthcare," *Future Generation Computer System*, vol. 78, pp. 659–676, 2018. *Crossref*, https://doi.org/10.1016/j.future.2017.04.036

[3] S. Aiswarya et al., "IoT based Big data Analytics in Healthcare: A Survey," *Proceedings of the First International Conference on Advanced Scientific Innovation in Science, Engineering and Technology, ICASISET 2020. Crossref,* https://doi.org/10.4108/eai.16-5-2020.2304020

[4] A. Paul et.al., "Fog Computing-Based IoT for Health Monitoring System," *Journal of Sensors*, vol. 2018, 2018. *Crossref*, https://doi.org/10.1155/2018/1386470

[5] Laila Fetjah, et al., "Towards a Smart Healthcare System: an Architecture Based on IoT, Blockchain, and Fog Computing," *International Journal of Healthcare Information Systems and Informatics (IJHISI)*, vol. 16, no. 4, pp. 1-18. 2021. *Crossref*, https://doi.org/10.4018/IJHISI.20211001.oa16

[6] Jean Bartra Lujan et al., "Telemedicine Prototype to Improve Medical Care and Patient and Physician Safety in Lima-Peru," *International Journal of Engineering Trends and Technology*, vol. 70, no. 8, pp. 83-96, 2022. *Crossref*, https://doi.org/10.14445/22315381/IJETT-V70I8P208

[7] Li G et al., "Service Popularity-Based Smart Resources Partitioning for Fog Computing-Enabled Industrial Internet of Things," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4702–4711, 2018. *Crossref*, https://doi.org/10.1109/TII.2018.2845844

[8] Alam MGR, Tun YK, and Hong CS, "Multi-Agent and Reinforcement Learning Based Code Offloading in Mobile Fog," *International Conference on Information Networking (ICOIN)*, pp. 285-290, 2016. *Crossref*, https://doi.org/10.1109/ICOIN.2016.7427078

[9] Kao Y-H et al., "Hermes: Latency Optimal Task Assignment for Resource Constrained Mobile Computing," *IEEE Transactions on Mobile Computing*, vol. 16, no. 11, pp. 3056–3069, 2017. *Crossref*, https://doi.org/10.1109/TMC.2017.2679712

[10] Nishtala R et al., "Hipster: Hybrid Task Manager for Latency-Critical Cloud Workloads," *2017 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pp. 409-420, 2017. *Crossref*, https://doi.org/10.1109/HPCA.2017.13

[11] Naas MI et al., "iFogStor: an IoT data placement Strategy for Fog Infrastructure," *2017 IEEE 1st International Conference on Fog and Edge Computing (ICFEC)*, pp. 97-104, 2017. *Crossref*, https://doi.org/10.1109/ICFEC.2017.15

[12] Pan J, and McElhannon J, "Future Edge Cloud and Edge Computing for Internet of Things Applications," *IEEE Internet of Things Journal*, vol. 5, no.1, pp. 439–49, 2017. *Crossref*, https://doi.org/10.1109/JIOT.2017.2767608

[13] Cao H, and Cai J, "Distributed Multiuser Computation Offloading for Cloudlet-Based Mobile Cloud Computing: A Game-Theoretic Machine Learning Approach," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 1, pp. 752–64, 2017. *Crossref*, https://doi.org/10.1109/TVT.2017.2740724

[14] Brogi A, and Forti S, "QoS-Aware Deployment of IoT Applications Through the Fog," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1185–1192, 2017. *Crossref*, https://doi.org/10.1109/JIOT.2017.2701408

[15] Mahmud R, Koch FL, and Buyya R, "Cloud-Fog Interoperability in IoT-Enabled Healthcare Solutions," *Proceedings of the 19th International Conference on Distributed Computing and Networking*, 2018. *Crossref*, https://doi.org/10.1145/3154273.3154347

[16] Rafique H. et al., "A Novel Bio-Inspired Hybrid Algorithm (NBIHA) for Efficient Resource Management in Fog Computing," *IEEE Access*, vol. 7, pp. 115760–115773, 2019. *Crossref*, https://doi.org/10.1109/ACCESS.2019.2924958

[17] Kishor A, Chakraborty C, and Jeberson W, "A Novel Fog Computing Approach for Minimization of Latency in Healthcare Using Machine Learning," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 6, no. 7, pp. 7-17, 2021. *Crossref*, https://doi.org/10.9781/ijimai.2020.12.004

[18] Sitanaboina S L Parvathi, and Harikiran Jonnadula, "A Hybrid Semantic Model for MRI Kidney Object Segmentation with Stochastic Features and Edge Detection Techniques," *International Journal of Engineering Trends and Technology*, vol. 70, no. 9, pp. 411-420, 2022. *Crossref*, https://doi.org/10.14445/22315381/IJETT-V70I9P242

[19] Aiswarya S et al., “Internet of Health Things: A Fog Computing Paradigm,” *2022 6th International Conference on Trends in Electronics and Informatics (ICOEI),* pp. 598-604, 2022. *Crossref,* https://doi.org/10.1109/ICOEI53556.2022.9776777

[20] Ngoc-Thanh Dinh and Younghan Kim, “An Efficient Availability Guaranteed Deployment Scheme for IoT Service Chains Over Fog-Core Cloud Networks,” *Sensors,* vol. 18, no. 11, pp. 3970. 2018. *Crossref*, https://doi.org/10.3390/s18113970

[21] Ahsan et al., "A Fog-Centric Secure Cloud Storage Scheme," *IEEE Transactions on Sustainable Computing,* vol. 7, no. 2, pp. 250-262, 2019. *Crossref,* https://doi.org/10.1109/TSUSC.2019.2914954

[22] Inas Bellary, and Vaishali Bagade, "Survey on Iot Based Architecture in Healthcare Applications," *SSRG International Journal of Electronics and Communication Engineering*, vol. 6, no. 5, pp. 1-5, 2019. *Crossref,* https://doi.org/10.14445/23488549/IJECE-V6I5P101

[23] Pedro H.Vilela et al., "Performance evaluation of a Fog-assisted IoT solution for e-Health applications," *Future Generation Computer Systems,* vol. 97, pp. 379-386, 2019. *Crossref*, https://doi.org/10.1016/j.future.2019.02.055

[24] Kakunuri Sreelatha et.al., “Integrity and Memory Consumption Aware Electronic Health Record Handling in Cloud,” *Concurrent Engineering,* vol. 29, no. 3 pp. 258-265. 2021. *Crossref,* https://doi.org/10.1177/1063293X211027869

[25] Sushmitha J et al., "Patient Medical Checkup using Webapp and IOT," *SSRG International Journal of Computer Science and Engineering*, vol. 5, no. 8, pp. 15-18, 2018. *Crossref,* https://doi.org/10.14445/23488387/IJCSE-V5I8P104

[26] Venkatesan C et al., “ECG Signal Pre-Processing and SVM Classifier-Based Abnormality Detection in Remote Healthcare Applications,” *IEEE Access,* vol. 6, pp. 9767-9773, 2018. *Crossref,* https://doi.org/10.1109/ACCESS.2018.2794346

[27] Harimoorthy K, and Thangavelu M, “Multi-Disease Prediction Model Using Improved SVM-Radial Bias Technique in Healthcare Monitoring System,” *Journal of Ambient Intelligence and Humanized Computing,* vol. 12, pp. 3715-3723, 2021. *Crossref,* https://doi.org/10.1007/s12652-019-01652-0

[28] Sakil Ahammed et al., “An IoT-based Real-Time Remote Health Monitoring System,” *International Journal of Recent Engineering Science,* vol. 8, no. 3, pp. 23-29, 2021. *Crossref,* https://doi.org/10.14445/23497157/IJRES-V8I3P104

[29] Rajkumar Buyya, and Satish Narayana Srirama, "Modeling and Simulation of Fog and Edge Computing Environments Using iFogSim Toolkit," *Fog and Edge Computing: Principles and Paradigms*, pp. 433-465, 2019. *Crossref,* https://doi.org/10.1002/9781119525080.ch17

[30] Kamruzzaman M. M et al., "Fuzzy-Assisted Machine Learning Framework for the Fog-Computing System in Remote Healthcare Monitoring," *Measurement*, vol. 195, 2022. *Crossref,* https://doi.org/10.1016/j.measurement.2022.111085

[31] Matthew N.O.Sadiku, Shumon Alam, and Sarhan M.Musa, "IOT for Healthcare," *SSRG International Journal of Electronics and Communication Engineering,* vol. 5, no. 11, pp. 1-5, 2018. *Crossref*, https://doi.org/10.14445/23488549/IJECE-V5I11P102

[32] S. Aiswarya et al., "A Time Optimization Model for the Internet of Things-Based Healthcare System Using Fog Computing," *2021 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES),* pp. 1-6, 2021. *Crossref,* https://doi.org/10.1109/ICSES52305.2021.9633874

[33] Shukla S et al., "A 3-Tier Architecture for Network Latency Reduction in Healthcare Internet-of-Things Using Fog Computing and Machine Learning," *8th International Conference on Software and Computer Applications,* pp. 522-528, 2019. *Crossref,* https://doi.org/10.1145/3316615.3318222

[34] Quang Duy La et al., "Enabling Intelligence in Fog Computing to Achieve Energy and Latency Reduction," *Digital Communications and Networks,* vol. 5, no. 1, Pp. 3-9, 2019. *Crossref,* https://doi.org/10.1016/j.dcan.2018.10.008

[35] Shukla S et al., “Architecture For Latency Reduction in Healthcare Internet-Of-Things Using Reinforcement Learning and Fuzzy Based Fog Computing,” *3rd International Conference of Reliable Information and Communication Technology*, pp. 372-383, 2018. *Crossref,* https://doi.org/10.1007/978-3-319-99007-1_36

[36] Singh et al., "Internet of Things for Sustaining a Smart and Secure Healthcare System," *Sustainable Computing: Informatics and Systems,* vol. 33, no. 1, 2022. *Crossref,* https://doi.org/10.1016/j.suscom.2021.100622