

Adaptive AI-Driven Multi-Objective IoT–Edge–Cloud Resource Optimization for QoS-Aware Telemedicine

Masood Ahmad*

Department of Computer Science, Abdul Wali Khan University Mardan. Corresponding
Author Email: masood@awkum.edu.pk

Ishtiaq Wahid

Department of Computer Science, Abdul Wali Khan University Mardan.

Arshad Ali

Directorate of IT, Abdul Wali Khan University Mardan,

Ihtiram Ullah

Department of Computer Science, Abdul Wali Khan University Mardan.

Abstract

Author Details

Received on 26 Dec, 2025

Accepted on 27 Jan, 2026

Published on 30 Jan, 2026

Corresponding E-mails & Authors*:

Masood Ahmad

masood@awkum.edu.pk

The incorporation of the Internet of Things (IoT), edge computing, and cloud computing has opened up new possibilities for offering intelligent and uninterrupted telemedicine services. Wearable sensors, medical monitoring tools, smart gateways, and connected medical facilities produce various patient data in real-time, which necessitates immediate processing and reliable communication. Nevertheless, the traditional telemedicine infrastructure that relies heavily on cloud computing is

faced with several issues such as delays, excessive energy consumption from communication, poor utilization of resources, and increased costs of cloud computing among other problems that may emerge from variability in healthcare workloads. The research involves the development of an adaptive multi-objective resource optimization system based on AI technology, which can be used in telemedicine solutions based on IoT, Edge, and Cloud technology. The proposed model includes healthcare IoT systems, cloud technology, edge nodes, prediction algorithms, classification of healthcare workloads, and the Adaptive Honey Bee Multi-Objective Optimization algorithm. Predictive algorithms determine potential future workloads in healthcare setting and provide timely resource allocation, while the optimization algorithm helps determine the optimal placement of tasks and resource allocation. Thus, it becomes possible to solve the problems of minimizing costs, delays, energy consumption, and SLA violations while maximizing resource utilization. The system assigns different priorities to tasks

depending on their clinical urgency ensuring that tasks such as emergency ECG and fall detection are processed first. The newly developed system creates a unified approach for managing various healthcare requirements. A methodology for evaluating the effectiveness of the AHBMO system is also proposed.

Keywords: Internet of Things, telemedicine, edge computing, cloud computing, multi-objective optimization, artificial intelligence, healthcare IoT, Honey Bee Optimization, resource allocation, QoS.

I. Introduction

Telemedicine has turned out to be a fundamental aspect of contemporary medicine as it allows for patients' receiving medical consultations, monitoring, diagnosis, and follow-up services without having to visit health facilities constantly. With the rapid evolution of the Internet of Things (IoT), telemedicine has been further improved, as a result of which physiological and contextual data is collected continuously with the use of connected sensors and medical devices. Wearable devices, such as ECG sensors, blood-pressure monitors, oximeters, glucose monitoring systems, temperature sensors, as well as other devices, make it possible to receive patient health information continuously and thus detect abnormal conditions early as well as increase the efficiency of health services [1]. Despite the potential benefits of IoT-based telemedicine, it brings various processing and transmission challenges. Healthcare sensors produce heterogeneous data at distinctive rates and with diverse quality requirements. Take, for example, an ECG sensor that produces a continuous stream of data that has to be processed in a hurry, while historical patient data can tolerate greater processing delays. An emergency fall-detection event has to be processed immediately, while regular activity data can get through with less frequency [2].

Traditional telemedicine architecture uses centralized cloud computing for data processing and storing. Cloud computing provides with the ability to leverage a lot of computational capacities, which is very beneficial for complicated health analytics, medical image processing, and needs of training AI models. However, sending all the data generated by IoT to cloud infrastructure may be expensive due to delays in communication, network flow, energy consumption of the IoT devices, and expenses associated with them [3].

Edge computing presents a practical approach to the issue of making computing resources readily available in close proximity to both patients and IoT devices. Healthcare systems do not need to send all patient information to reside in distant cloud systems; rather, only specific healthcare applications can be performed at locations of the edge servers or gateways [4].

Therefore, ECG incidents, materialization of falls, patient monitoring, etc., can be done at places not too far from the patients, while tasks requiring a lot of computing resources but being less urgent can be processed at the cloud servers [5].

Despite the fact that introduction of edge computing has been a solution to the problem, it does not mean that the issue of managing resources has been solved. Edge nodes have naturally less computing power than centralized cloud servers. Moreover, healthcare tasks can change in time and what is true today from the point of view of the number of patients and amount of sensor information required for video consultations can turn to be false tomorrow. Static allocation of resources can lead to overloading some nodes and underutilization of the others [6].

AI can remedy the situation and help anticipate the workload in advance by examining the previous workload information and remotely predicting CPU requirements, necessary memory, network traffic, and the number of active healthcare tasks. However, resource allocation problems in IoT-edge-cloud telemedicine are multi-objective optimization questions, since lowering latency may require higher dependency on edge devices, which, in turn requires more energy [7].

To tackle the problems posed by these challenges, the current paper presents an Adaptive AI-Driven Multi-Objective IoT-Edge-Cloud Resource Optimization Framework for QoS-Aware Telemedicine. This framework consists of features such as AI-based workload prediction, healthcare-aware QoS classification, adaptive data transmission, dynamic task placement, and Adaptive Honey Bee Multi-Objective Optimization. Major contributions of this study involve the following:

- A combined IoT-Edge-Cloud architecture is proposed to provide smart and resource-efficient telemedicine services.
- The AI-based workload prediction system is used to estimate future loads in the domain of healthcare and plan resources proactively.
- QoS-driven healthcare task classification system is also developed to identify different tasks based on latency, bandwidth, and computational power.
- A multi-objective optimization model is developed to minimize latency, energy consumption, cloud costs, and SLA violations maximally.
- The Adaptive Honey Bee Multi-Objective Optimization (AHBMO) mechanism is presented for dynamic task placement. The adaptive IoT communication system is integrated to provide for timely communication of important events and reduce unused transmissions.

The remainder of the paper is organized as follows. Section II discusses related work. Section III presents the proposed IoT-Edge-Cloud architecture. Section IV

formulates the system model and optimization problem. Section V presents the proposed AHBMO methodology. Section VI describes the experimental methodology. Section VII discusses the expected performance and research implications. Section VIII concludes the paper.

2. Related work

2.1 IoT-Enabled Telemedicine

Healthcare IoT has made it easier to monitor patients constantly using various sensors and medical devices. The healthcare IoT systems create physiological measures that allow these measures to be sent to healthcare systems for decision-making. Such systems are very useful for the elderly, chronic disease management, long-distance patient monitoring, home healthcare, and emergency alert detection [8].

However, healthcare IoT devices are very limited regarding their battery capacity, processing power, memory, and communication speed. Thus, the continuous transmission of raw sensor data uses a lot of energy and creates unnecessary data flow on the networks [9].

Besides, different healthcare applications have different Quality of Service requirements. Therefore, while for critical events quick response is necessary, for regular events there can be more time allowed for the processing of the preventive measures [10].

2.2. Edge Computing in Telemedicine

Edge computing allows to decrease the distance between data source and data processing by locating computing elements in proximity to patients. This approach helps to eliminate communication delays and congestions [11].

In this situation, abnormal ECG signal can be examined at the edge node, which will automatically generate data on detected problem. However, all historical ECG values will be sent to the cloud to be processed using long-term analytics [12].

The most advanced part is that despite the fact that edge computing is aimed at the reduction of delays, edge resources are still limited. The ineffective placement of the tasks can overload edge servers with tasks while underutilizing others. That's why dynamic resource allocation is of great importance for efficient functioning of the IoT-Edge-Cloud healthcare systems [13].

2.3. Cloud Resource Management

Cloud computing offers healthcare applications flexible computational and storage resources. With dynamic provisioning functionality, the cloud infrastructure can adjust according to variable workloads. However, over-provisioning may lead to higher prices, while under-provisioning may cause high latency and SLA violations [14].

Most existing methods concentrate on one specific goal, such as cost or latency. However, telemedicine solutions require simultaneous optimization of multiple conflicting objectives [15].

2.4. AI-Based Resource Management

Application of the concept of machine learning has been utilized in various areas such as workload forecasting, resource-demand forecasting, anomaly detection, and intelligent scheduling. Workload forecasting helps cloud and edge computing platforms provision resources ahead of time. [16]

Yet, prediction itself does not provide a solution for optimal allocation of resources of different types. Therefore, advanced optimization methods such as Genetic Algorithm, Particle Swarm Optimization, Ant Colony Optimization, and Honey Bee Optimization have been studied for the allocation problem [17]. Algorithms with a multiple objective function, namely NSGA-II and MOPSO algorithms, allow achieving multiple Pareto optimal solutions. But still, there is a need for an integrated method that will combine workload forecasting based on AI, healthcare QoS, IoT energy limitations, edge cloud placement, and adaptive multi-objective optimization [18]. The proposed methodology closes this gap in research by introducing the mentioned components as a single telemedicine resources management system [19].

3. Proposed IoT–edge–cloud telemedicine framework

3.1. System Architecture

The proposed framework consists of five major layers:

1. Healthcare IoT Layer
2. IoT Gateway and Communication Layer
3. Edge Computing Layer
4. Cloud Computing Layer
5. Intelligent Resource Management Layer

The overall data flow is:

Patient → IoT Sensors → IoT Gateway → Edge/Cloud → Telemedicine Services → Healthcare Provider

The intelligent resource management layer works within these components, acting in real-time to decide how the tasks in healthcare should be sent, processed, and distributed.

3.1.1 Healthcare IoT Layer

The IoT layer consists of variously designed sensors and medical devices working in contact with patients. Let the IoT sensors be represented as follows.

$$S = \{s_1, s_2, \dots, s_n\}$$

Each sensor generates healthcare information at a particular rate and has specific energy, bandwidth, latency, and priority requirements.

A sensor can be represented as:

$$s_i = \{D_i, R_i, E_i, L_i, P_i\}$$

where:

- D_i = data generation rate of sensor i
- R_i = computational/resource requirement
- E_i = energy characteristics
- L_i = latency requirement
- P_i = clinical priority

Important IoT devices include ECG detectors, SPO2 detectors, devices for measuring blood pressure and glucose, temperature meters, motion detectors, systems for detecting falls, and smart cameras. The various characteristics of these devices lead to differing needs for computing and communications.

3.1.2 IoT Gateway Layer

IoT gateways allow for the connection of sensors and edge infrastructures. The gateway assures that data aggregation, data filtering, data compression, protocol conversion, first-order anomaly detection, and transmission scheduling occur. It also acts as a launching pad for adaptive data transmission where only crucial data is transmitted. However, this rule does not prevent important healthcare events from adhering to the same scheme.

3.1.3 Edge Computing Layer

The edge layer contains computational nodes located close to patients.

Let the set of edge nodes be:

$$E = \{e_1, e_2, \dots, e_m\}$$

All edge nodes offer CPU, memory, storage, and communication capabilities. The edge layer is mainly intended for latency-sensitive processes, for example: Emergency detection, ECG anomaly detection, Fall detection, Real-time physiological analysis, and Interactive telemedicine sessions. Local processing of workload allows for significant reduction of communication delays.

3.1.4 Cloud Computing Layer

The cloud layer provides large-scale computational and storage resources. Let the set of cloud resources be:

$$C = \{c_1, c_2, \dots, c_k\}$$

Cloud resources are ideal for: Medical image processing, AI model training, Long-term data storage, Historical data analysis, Population-level health data analysis, Large video

processing and Difficult diagnostics. The optimizer chooses if the task must be completed at the edge or cloud resource.

3.2. System Model And Problem Formulation

3.2.1 Healthcare Task Model

Let the set of healthcare tasks generated by IoT devices be:

$$T = \{T_1, T_2, \dots, T_n\}$$

Each healthcare task is represented by:

$$T_i = \{CPU_i, MEM_i, BW_i, L_i, P_i\}$$

where:

- CPU_i = CPU requirement
- MEM_i = memory requirement
- BW_i = bandwidth requirement
- L_i = maximum tolerable latency
- P_i = clinical priority

Tasks are divided into different categories based on their level of urgency.

Critical Tasks: Examples of critical tasks may pertain to Emergency ECG, Severe Hypoxemia, Fall Detection, Lifethreatening abnormality, etc.

High-Priority Tasks: Examples of high-priority tasks include Real-time monitoring of the patient, Video consultation, Abnormal blood pressure detection, etc.

Medium-Priority Tasks: Few examples of medium-priority tasks are Routine physiological monitoring, Periodical monitoring of the patients, etc.

Low-Priority Tasks: Some of the low-priority processes include Historical analytics, Long-term data processing, Non-urgency reporting, etc.

3.2.2 Resource Model

The available computational resources are represented as:

$$R = R_{IoT} \cup R_{edge} \cup R_{cloud}$$

Each resource node j is represented as:

$$R_j = \{CPU_j, MEM_j, BW_j, E_j\}$$

where:

- CPU_j = CPU capacity
- MEM_j = memory capacity
- BW_j = bandwidth capacity
- E_j = energy characteristics

3.2.3 Task Placement

A binary task-placement variable is defined as:

$$x_{ij} = 1, \text{ if task } i \text{ is assigned to resource } j$$

$$x_{ij} = 0, \text{ otherwise}$$

Every task must be assigned to exactly one valid processing resource:

$$\sum_{j=1}^J x_{ij} = 1$$

The CPU capacity constraint is:

$$\sum_{i=1}^N x_{ij} CPU_i \leq CPU_j$$

The memory constraint is:

$$\sum_{i=1}^N x_{ij} MEM_i \leq MEM_j$$

The bandwidth constraint is:

$$\sum_{i=1}^N x_{ij} BW_i \leq BW_j$$

The binary placement constraint is:

$$x_{ij} \in \{0, 1\}$$

These constraints ensure that a resource cannot be assigned more workload than its available capacity.

3.3. Energy Consumption Model

3.3.1 IoT Communication Energy

Communication energy is particularly important because many healthcare IoT devices operate using batteries. Let $s_i(t)$ indicate whether sensor i transmits data at time t . The communication energy is modeled as:

$$E_{comm(t)} = \sum_{i=1}^N s_i(t) D_i(t) e_i^{tx}$$

where:

- $D_i(t)$ = transmitted data volume
- e_i^{tx} = energy required to transmit one unit of data

The total communication energy is:

$$E_{IoT} = \sum_{t=1}^T E_{comm(t)}$$

Reducing unnecessary transmissions can therefore directly reduce IoT energy consumption.

3.3.2 Processing Energy

The processing energy consumed by edge and cloud resources is represented as:

$$E_{proc} = \sum_{j=1}^J \sum_{i=1}^N x_{ij} CPU_i P_j \tau_i$$

where:

- P_j = power consumption of resource j
- τ_i = processing time of task i

The total system energy is:

$$E_{total} = E_{IoT} + E_{proc}$$

The optimizer attempts to minimize this value while satisfying healthcare QoS constraints.

3.4 Latency Model

The end-to-end latency of a healthcare task consists of transmission, queuing, processing, and response delays:

$$L_i = L_i^{tx} + L_i^{q\ ueue} + L_i^{proc} + L_i^{resp}$$

where:

- L_i^{tx} = transmission latency
- $L_i^{q\ ueue}$ = queuing latency
- L_i^{proc} = processing latency
- L_i^{resp} = response latency

Average latency is:

$$L_{avg} = \left(\frac{1}{N}\right) \sum_{i=1}^N L_i$$

Each task must satisfy:

$$L_i \leq L_i^{max}$$

Critical healthcare tasks are therefore prevented from being assigned to resources that would cause unacceptable delay.

3.5 Cloud Cost Model

Cloud resource usage generates computational, storage, and communication costs. The cost associated with task i is:

$$C_i = C_i^{cpu} + C_i^{mem} + C_i^{storage} + C_i^{net}$$

where the individual terms represent CPU, memory, storage, and network costs. The total cloud cost is:

$$C_{total} = \sum_{i=1}^N C_i$$

The optimizer attempts to minimize cloud cost while preserving required QoS.

3.6 SLA Violation Model

A task is considered to violate its SLA when its QoS requirements are not satisfied. The SLA violation rate is:

$$V_{SLA} = \frac{N_{violated}}{N_{total}}$$

or, as a percentage:

$$V_{SLA(\%)} = \left(\frac{N_{violated}}{N_{total}}\right) \times 100$$

A low SLA violation rate is essential for reliable telemedicine services.

3.7. Multi-Objective Optimization Formulation

The proposed framework simultaneously considers five objectives.

Objective 1: Minimize latency

$$Minimize F_1 = L_{avg}$$

Objective 2: Minimize energy consumption

$$\text{Minimize } F_2 = E_{total}$$

Objective 3: Minimize cloud cost

$$\text{Minimize } F_3 = C_{total}$$

Objective 4: Minimize SLA violations

$$\text{Minimize } F_4 = V_{SLA}$$

Objective 5: Maximize resource utilization

Resource utilization is defined as:

$$U_{resource} = \left(\frac{\sum_{j=1}^J ResourceUsed_j}{\sum_{j=1}^J ResourceCapacity_j} \right) \times 100$$

Since the optimization framework minimizes objectives, the fifth objective can be expressed as:

$$\text{Minimize } F_5 = 1 - U_{resource}$$

The complete optimization problem is therefore:

$$\text{Minimize } F(X) = [F_1(X), F_2(X), F_3(X), F_4(X), F_5(X)]$$

subject to:

$$\begin{aligned} \sum_{j=1}^J x_{ij} &= 1 \\ \sum_{i=1}^N x_{ij} CPU_i &\leq CPU_j \\ \sum_{i=1}^N x_{ij} MEM_i &\leq MEM_j \\ \sum_{i=1}^N x_{ij} BW_i &\leq BW_j \\ L_i &\leq L_i^{max} \\ x_{ij} &\in \{0, 1\} \end{aligned}$$

This formulation represents the competing objectives involved in IoT–Edge–Cloud telemedicine resource management.

3.8 AI-Based Workload Prediction

3.8.1. Workload Representation

The proposed framework uses AI to predict future healthcare workloads. The observed workload at time t is represented as:

$$W(t) = \{CPU(t), MEM(t), BW(t), N_{task(t)}\}$$

where:

- $CPU(t)$ = CPU utilization
- $MEM(t)$ = memory utilization
- $BW(t)$ = network bandwidth utilization
- $N_{task}(t)$ = number of active tasks

The prediction model estimates the next workload:

$$W(t+1) = f(W(t), W(t-1), \dots, W(t-k))$$

where:

- $W(t+1)$ = predicted workload
- k = historical observation window
- $f(\cdot)$ = AI prediction model

A recurrent neural network such as LSTM or GRU can be used for this purpose.

3.8.2. Proactive Resource Allocation

The resource-management module is presented with the anticipated load prior to the next scheduling period. If the anticipated load is higher than the available capacity at the edge, then additional cloud resources can be deployed proactively. On the other hand, if a large drop in the workload is anticipated, unnecessary cloud resources can be removed. This approach helps to prevent both resource shortages and excess provisioning.

4. Proposed Adaptive Honey Bee Multi-Objective Optimization

4.1. Solution Representation

The AHBMO algorithm which has been suggested takes inspiration from the manner in which colonies of honey bees hunt for food. Each of the food candidates will be one of the solutions on allocation of resources.

$$X = [x_1, x_2, \dots, x_n]$$

where each element represents the selected processing resource for a healthcare task. For example:

$$X = [E_2, C_1, E_1, C_3, E_2, \dots]$$

indicates the processing location selected for each task.

4.2 Objective Normalization

Because the optimization objectives have different units, they are normalized.

For minimization objectives:

$$F_{k\text{norm}} = \frac{(F_k - F_{k\min})}{(F_{k\max} - F_{k\min} + \varepsilon)}$$

where ε is a small positive value that prevents division by zero.

The normalized values are used during solution evaluation.

4.3. Adaptive Fitness

An adaptive fitness function can be expressed as:

$$\text{Fitness}(X) = \sum_{k=1}^K w_k F_{k\text{norm}}(X)$$

where:

- w_k = weight of objective k
- K = number of objectives

Nevertheless, due to the different priorities of healthcare applications, it is possible to change the weights in real-time, in response to workload conditions and urgency of clinical cases, thus assigning more importance to aspects like latency and SLA compliance in scenarios of emergency workload.

4.4. Adaptive Exploration And Exploitation

A key component of AHBMO is adaptive control of exploration and exploitation.

The exploration coefficient is defined as:

$$\alpha(t) = \alpha_{min} + (\alpha_{max} - \alpha_{min}) \left(1 - \frac{t}{T_{max}}\right)$$

where:

- $\alpha(t)$ = exploration coefficient
- α_{min} = minimum exploration value
- α_{max} = maximum exploration value
- t = current iteration
- T_{max} = maximum number of iterations

In the beginning, when optimization starts $\alpha(t)$ is set very high and will try to explore as many configurations as possible. But with the progress of optimization, $\alpha(t)$ decreases and it focuses on the solution that showed good behavior and thus makes exploitation prevail.

$$\alpha'(t) = \alpha(t)(1 + \lambda V_w)$$

where:

- V_w = workload variation
- λ = workload sensitivity coefficient

A rapidly changing workload therefore encourages additional exploration.

4.5. QoS-Aware Optimization

Healthcare tasks cannot be treated equally because their clinical importance varies.

A QoS penalty is introduced as:

$$P_{QoS} = \sum_{i=1}^N P_i \max(0, L_i - L_i^{max})$$

Here, P_i stands for the clinical importance of task i . The mechanism guarantees that not complying with a time requirement for an urgent task leads to a much bigger penalty than disregarding the requirements of a low priority task. The QoS-aware optimization thus guarantees that the allocation of computing resources is in line with healthcare requirements.

4.6. Adaptive Task Placement

For each task, the optimizer determines whether it should be processed at:

$$P_i \in \{IoT, Edge, Cloud\}$$

The decision is based on: Latency requirements, CPU requirements, memory requirements, bandwidth availability, the current state of resources, energy use, costs, cloud computing, and importance in healthcare. "Critical tasks are assigned with preference to the nearby edge resources when enough resources are available. Computation-heavy operations can go to the cloud resources in the case of lacking capacity of the edge resources. Routine tasks can be delayed or summed when appropriate.

4.7. Adaptive IoT Data Transmission

Consistent transmission of every reading from each sensor will use too much energy for the IoT. As a result, the framework utilizes an adaptive transmission approach. The decision on whether to relieve the sensor is made based on:

$$s_i(t) = 1, \text{ if } A_i(t) \geq \theta_i$$

$$s_i(t) = 0, \text{ if } A_i(t) < \theta_i$$

where:

- $A_i(t)$ = significance of the new sensor observation
- θ_i = sensor-specific transmission threshold

The significance of the transmission can be determined by: Comparison with the last measurement obtained, Individual threshold for every patient, Indication of abnormalities, Clinical priority, Time repetition and Current state of the network. For example, when the heart rate of a patient is stable within normal values, several similar readings can be collected before sending. In case of sudden appearance of an abnormal ECG, the system will send the reading.

Algorithm 1: AHBMO for IoT–Edge–Cloud Resource Allocation

Input: Healthcare tasks T , IoT resources, edge resources, cloud resources, maximum iterations $T_{\max}$

Output: Pareto-optimal resource allocation solutions

1. Initialize a population of candidate resource-allocation solutions.
2. Evaluate CPU, memory, bandwidth, latency, energy, cost, and QoS constraints.
3. Calculate the objective values of each solution.
4. Identify non-dominated solutions.
5. Initialize the Pareto archive.
6. Set the initial exploration coefficient α .
7. Repeat until the maximum number of iterations is reached:
 - a. Select promising solutions from the Pareto archive.
 - b. Generate neighboring solutions around selected food sources.
 - c. Perform exploration using employed and scout bee operations.

- d. Perform exploitation around high-quality solutions.
- e. Apply healthcare QoS constraints.
- f. Reject infeasible resource allocations.
- g. Calculate latency, energy, cost, SLA violation, and utilization.
- h. Update the Pareto archive.
- i. Calculate workload variation.
- j. Adapt the exploration coefficient.
- k. Replace weak solutions with newly generated solutions.
- 8. Return the final Pareto archive.

The final archive contains alternative resource-allocation configurations representing different trade-offs among latency, energy consumption, cost, QoS, and utilization.

5. Experimental Evaluation

5.1. Simulation Environment

The proposed framework was developed in Python using a simulated telemedicine environment consisting of IoT-Edge-Cloud communications. The telemedicine environment will consist of: Multiple simulated patients; Heterogeneous IoT sensors; IoT gateways; Edge servers; Cloud servers; Dynamic healthcare workloads; Network bandwidth constraints; CPU and memory constraints; and Healthcare quality of service. Each individual simulated patient can be monitored with multiple IoT sensors. To illustrate, one patient may generate ECG, SpO₂, blood pressure, temperature, and activity data at the same time.

5.2. Healthcare Workload Model

The experimental workload will include several representative telemedicine services as shown in table 1.

Table 1: *Workload Intensity*

Service	CPU Demand	Bandwidth	Priority	Latency
Emergency ECG	High	High	Critical	Very Low
Fall Detection	Medium	Medium	Critical	Very Low
Video Consultation	High	Very High	High	Low
SpO ₂ Monitoring	Low	Medium	High	Low
Blood Pressure	Low	Low	Medium	Moderate
Temperature Monitoring	Low	Low	Medium	Moderate
Historical Analytics	High	Medium	Low	High
Medical Imaging	Very High	High	High	Moderate

The workload intensity will be varied to test the behavior of the proposed method under light, moderate, heavy, and highly dynamic conditions.

5.3. Baseline algorithms

The AHBMO method to be proposed will be compared to other current approaches such as:

Genetic Algorithm (GA): This method involves the search for solutions regarding resource allocation by applying selection, crossover, and mutation methods.

Particle Swarm Optimization (PSO): In PSO, candidate solutions are modelled as particles in the swarm method and the position of the particle changes based on the particle's previous experience and experience of all other particles.

NSGA-II: NSGA-II is a well-known multi-objective evolutionary optimization method which uses non-dominated sorting and crowding distance.

MOPSO: MOPSO method extends Particle Swarm Optimization for considering multi-objective problems by employing an external Pareto archive.

Standard Honey Bee Optimization: The ordinary honey bee optimization method will be used to understand the impact of the adaptive method on the performance of the proposed approach. All methods will operate with similar sizes of populations and limits of generations.

5.4 Performance Evaluation Metrics

Average Latency: Average latency is calculated as: $L_{avg} = \left(\frac{1}{N}\right) \sum_{i=1}^N L_i$. Lower latency indicates better responsiveness.

Energy Consumption: Total energy is: $E_{total} = E_{IoT} + E_{edge} + E_{cloud}$. Lower energy consumption indicates greater efficiency.

Cloud Cost: Total cloud expenditure is measured using computational, storage, and network resource consumption. Lower cost represents more economical resource utilization.

SLA Violation Rate: $V_{SLA(\%)} = \left(\frac{N_{violated}}{N_{total}}\right) \times 100$. Lower SLA violation is desirable.

Resource Utilization: $U(\%) = \left(\frac{ResourceUsed}{ResourceCapacity}\right) \times 100$. Higher balanced utilization indicates efficient use of available infrastructure.

Throughput: $Throughput = \frac{N_{success}}{T_{simulation}}$. where $N_{success}$ is the number of successfully completed tasks.

Hypervolume: Hypervolume is used to evaluate the quality of the Pareto front by considering both convergence and diversity. A larger hypervolume indicates a better set of Pareto-optimal solutions.

5.5. Results and Discussion

5.5.1 Experimental Configuration

In order to determine how useful the introduced Adaptive Honey Bee Multi-Objective Optimization (AHBMO) framework is, a simulation of the IoT-Edge-Cloud telemedicine system has been adopted. The simulation involves heterogeneous healthcare workloads from multiple patients benefitting from the internet of things. The simulated system consists of IoT gateways, edge server, and cloud facilities all with different computing powers.

Various approaches have been applied for the initial evaluation. These approaches include Genetic Algorithm (GA), Particle Swarm Optimization (PSO), NSGA-II, MOPSO, and AHBMO. These algorithms were used under equal simulation conditions. The configuration of simulation used to obtain the experimental data is presented in Table 2.

Table 2. *Simulation Configuration*

Parameter	Value	Parameter	Value
Number of patients	100–1000	Simulation duration	3600 s
IoT sensors per patient	5–8	Healthcare task types	8
IoT gateways	20	CPU capacity per edge node	8–32 cores
Edge servers	10	Memory per edge node	16–64 GB
Cloud servers	20	Cloud CPU capacity	32–64 cores
Optimization population	50	Network bandwidth	100–1000 Mbps
Maximum iterations	100	Optimization objectives	5

5.5.2 Convergence Analysis

One hundred iterations were performed in order to evaluate the performance of the optimization algorithms. The normalized multi-objective fitness value is defined in such a way that the smaller its value the better the solution. The values of convergence are presented in figure 1.

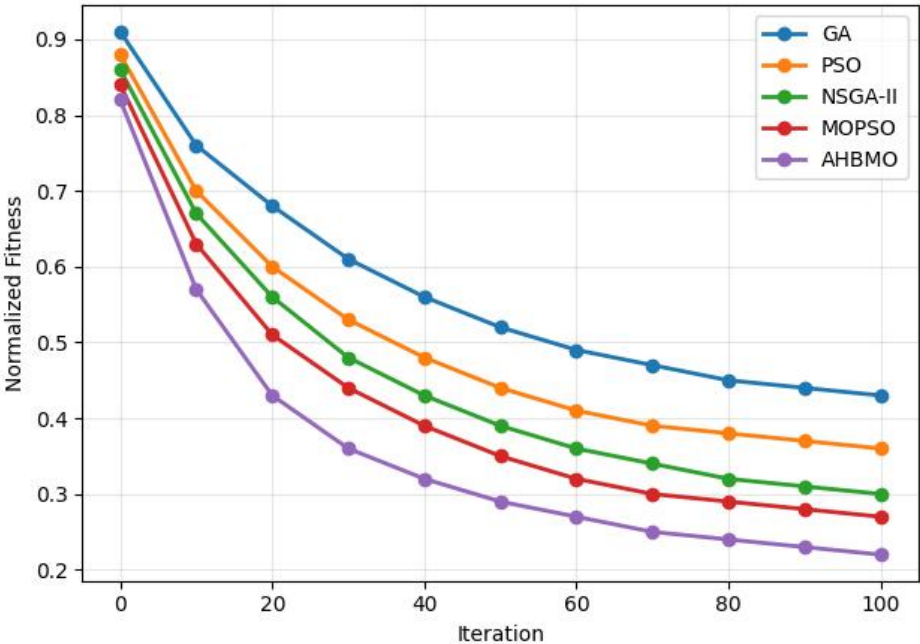


Figure 1. convergence comparisons of optimization algorithms

The results show that AHBMO reaches convergence at a higher speed than the other techniques. It should be remarked that during the 20th iteration AHBMO reaches a fitness value of 0.43 compared to GA (0.68), PSO (0.60), NSGA-II (0.56), and MOPSO (0.51). On the 100th iteration, the AHBMO shows a fitness value of 0.22. Such efficiency is due to the adaptive nature of the exploration-exploitation process. In effect, when optimization begins, the focus will be on most diverse resources, while in subsequent iterations the objective will be to ensure maximally successful movement through the available resources.

5.5.3 Average End-to-End Latency

Average end-to-end latency was evaluated under increasing healthcare workload. The number of active patients was varied from 100 to 1000.

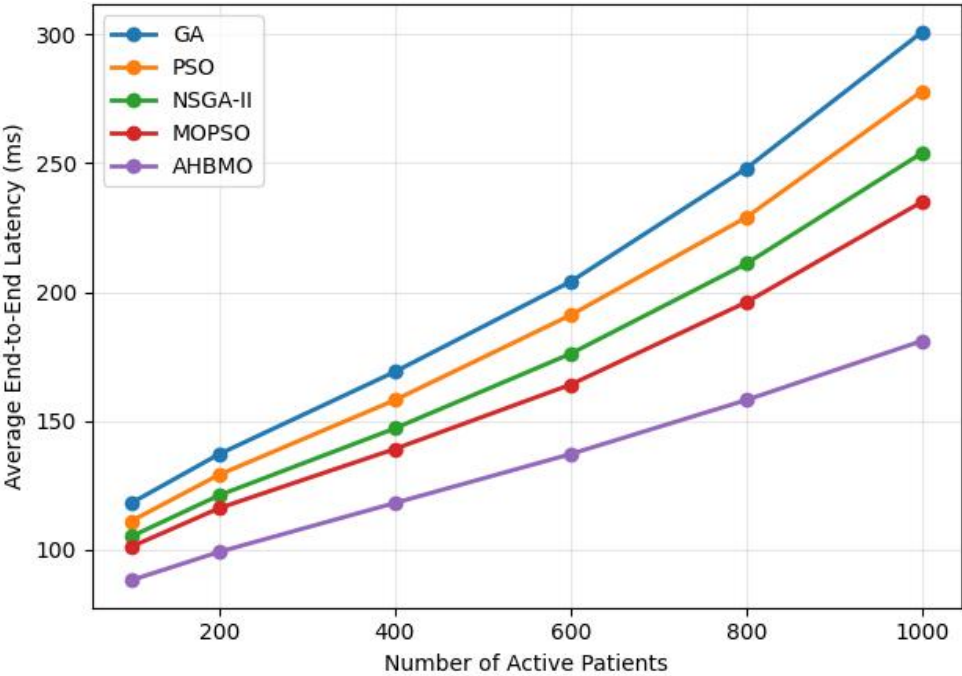


Figure 2. Average end-to-end latency under increasing workload

Figure 2 shows the relationship between the total patients and average end-to-end latency. AHBMO results in the lowest latency in all the workload conditions. When there are 1000 active patients, the latency of AHBMO is measured at 181ms, while other methods like GA, PSO, NSGA-II, and MOPSO return the latencies that are measured as 301ms, 278ms, 254ms, and 235ms, respectively. The proposed method, compared to GA, shows:

$$\text{Latency Reduction} = \left(\frac{(301 - 181)}{301} \right) \times 100 \approx 39.9\%$$

The efficiency of AHBMO over the other methods, like PSO, NSGA-II, and MOPSO, is, respectively: 34.9%, 28.7%, and 23.0%. This improvement can be attributed to the QoS-aware task-placement strategy developed in the process of implementing the current research. More important healthcare jobs and tasks are processed closer to the edge nodes available to ensure that the tasks are done on time with the minimum latency.

5.5.4. Energy Consumption

Energy efficiency is particularly important because IoT sensors are often battery powered. The total system energy consumption was evaluated under increasing workload.

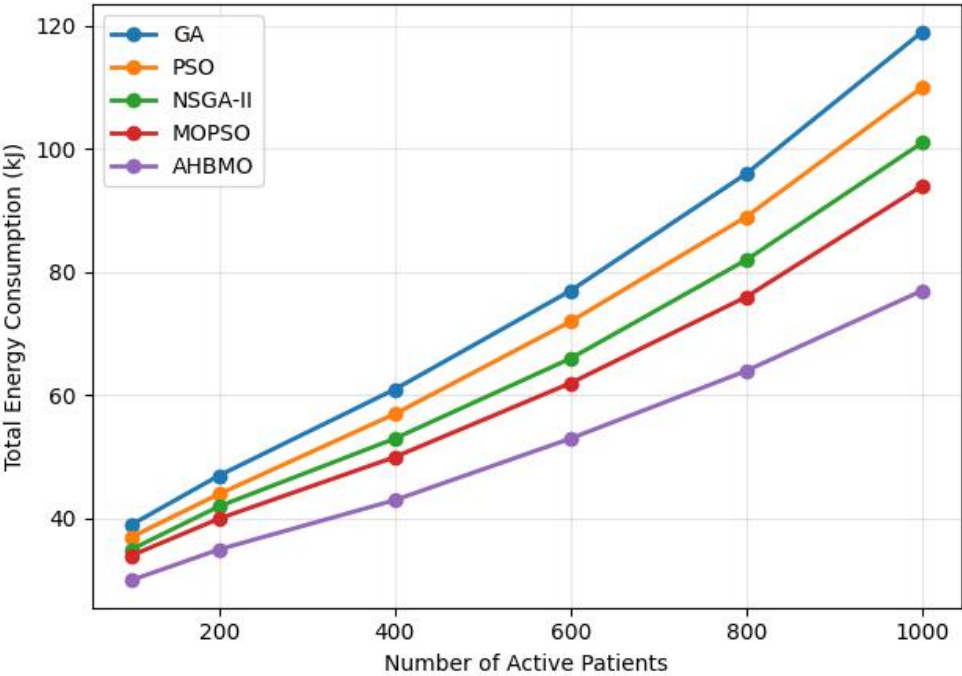


Figure 3. Energy consumption under increasing workload

The image in Figure 3 illustrates the energy consumption effect on the quantity of patients in an active situation. With the introduction of 1000 patients, the energy consumption in case of AHBMO is reported at 77 kJ, while in case of GA – at 119 kJ, which signifies a level of energy reduction of:

$$\text{Energy Reduction} = \left(\frac{(119 - 77)}{119} \right) \times 100 \approx 35.3\%$$

It is noteworthy that the suggested solution results in a significant drop in energy consumption, compared with the existing techniques. In this regard, a comparison made with the PSO, NSGA-II and the MOPSO shows that the method's energy usage is less than that of PSO and NSGA-II by 30.0% and 23.8%, correspondingly. The impact of this significant reduction can be explained by two main reasons. The first one is the fact that the adaptable transmission technique prevents the transmission of unneeded repeated measurements from sensors. The second reason is that the optimization technique attempts to provide the execution of tasks on resources that exhibit the most favorable energy-latency balance.

The outcomes obtained demonstrate that it is vital to take into consideration the energy spent for IoT communication together with that spent for processing in the cloud and on the edge.

5.5.5. Cloud Resource Cost

The effect of resource optimization on cloud operational cost was also investigated. The cost is expressed using normalized monetary units per simulation hour.

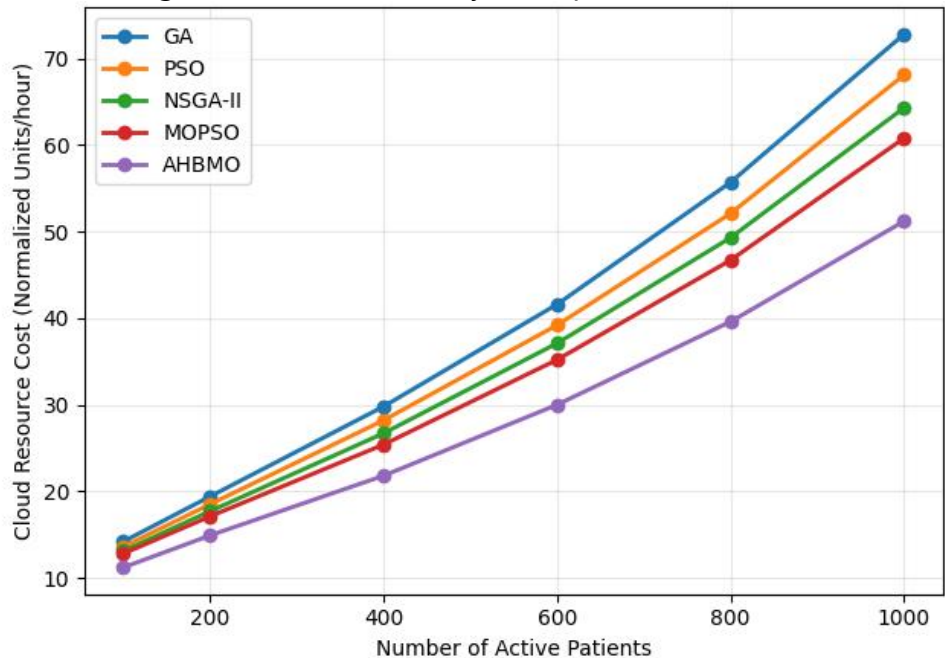


Figure 4. Cloud resource cost under increasing workload

The cost of cloud resources is shown in Figure 4 as a function of the number of patients actively using the service. With 1,000 patients, AHBMO has a cost associated with the cloud of 51.2 while the GA has a cost of 72.8. Hence, the cost reduction is calculated as follows:

$$\text{Cost Reduction} = \left(\frac{72.8 - 51.2}{72.8} \right) * 100 \approx 29.7\%$$

This effective framework makes the cost less through unnecessary provisioning of the cloud and making use of edge capacity. The AI workload predictor enables predictive scaling of the system which contributes to timely provision of resources before they reach their peak usage time. This fact outlines another advantage of the suggested approach IoT-Edge-Cloud. There are many health care tasks that could be conducted without using the cloud. Hence the intelligent placement of the workload allows reducing the cloud dependency and improving quality of service.

5.5.6 SLA Violation Rate

SLA compliance is particularly important in telemedicine because delayed healthcare services can negatively affect patient monitoring and clinical decision-making.

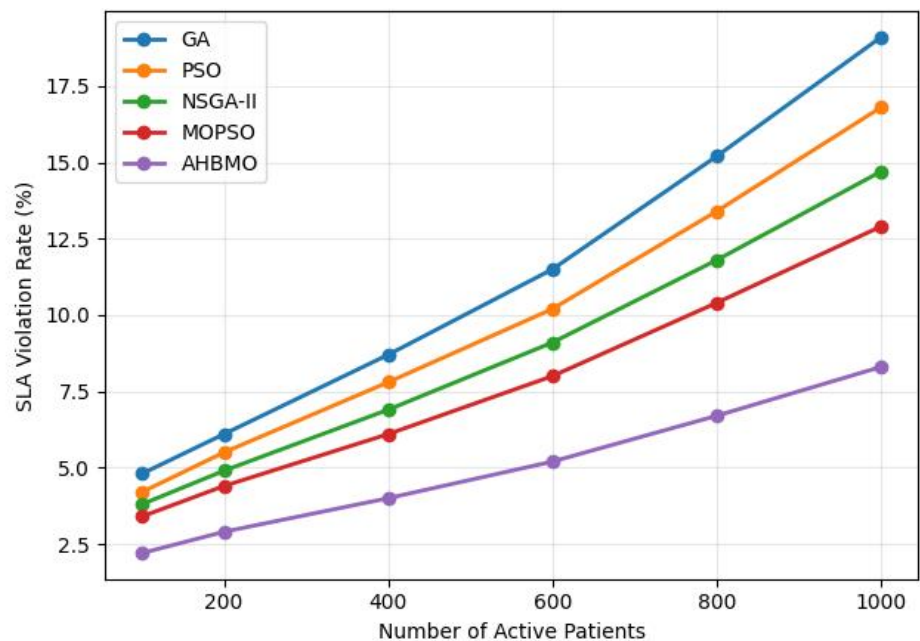


Figure 5. SLA violation rate

Figure 5 displays the SLA infringement rate as a workload function. With 1,000 patients, AHBMO has an SLA infringement rate of 8.3% while GA has 19.1%. The SLA improvement, when compared to GA, is as follows:

$$SLA\ Improvement = \left(\frac{(19.1 - 8.3)}{19.1} \right) \times 100 = 56.5\% .$$

The significance of the improvement is especially visible in medical mission-critical workloads. Due to the QoS-aware penalty system, any violations related to mission critical jobs incur greater penalties. As a result, the optimizer makes efforts to keep the edge capacity reserved for mission-critical jobs and prevent the allocation of delay-sensitive jobs to overloaded resources.

5.5.7 Resource Utilization

Resource utilization was evaluated to determine whether the proposed approach can effectively use available computational infrastructure.

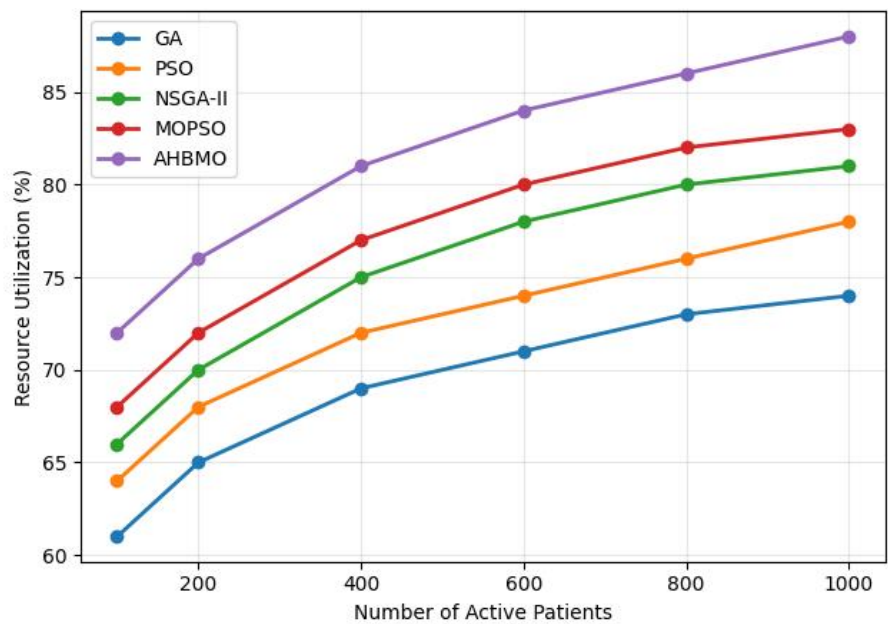


Figure 6. Resource Utilization

Resource utilization needs to be illustrated as a function of patient activity in Figure 6. Under the condition of 1000 patients, AHBMO manages an 88% utilization factor, compared to GA's 74%. Higher levels of utilization imply that the suggested approach is more successful in distributing workloads among the edge and cloud resources. Still, the objective is not achieving maximum utilization, as it may lead to queuing and delay.

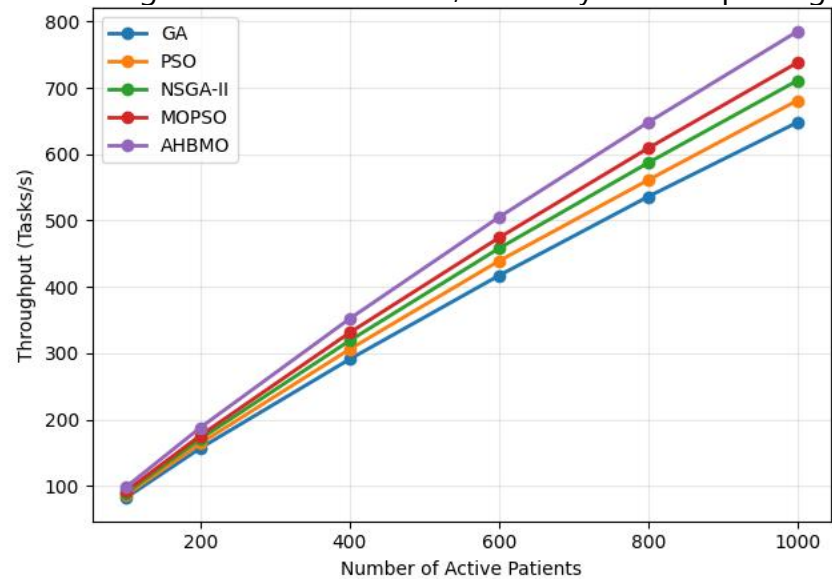


Figure 7. Throughput vs number of active patients

5.5.8 Throughput Analysis

Throughput represents the number of successfully processed healthcare tasks per second.

In Figure 7, throughput achievement is plotted against the number of patients. AHBMO achieved the throughput of 785 successfully executed tasks at 1000 patients compared to the rate of 648 tasks/sec achieved by GA. The improvement in throughput compared to GA can be calculated as follows:

$$\text{Throughput Improvement} = \left(\frac{(785 - 648)}{648} \right) \times 100 \approx 21.1\%$$

This suggests that an effective placement of tasks and efficient use of resources results in more healthcare processes being performed in a certain period of time.

5.5.9 Pareto-Front Analysis

Due to the presence of multiple objectives in the issue at hand, no one performance metric would be able to indicate the quality of the optimization. Hence, the Pareto front is also examined. To illustrate the situation, two objectives are chosen: energy consumption and latency.

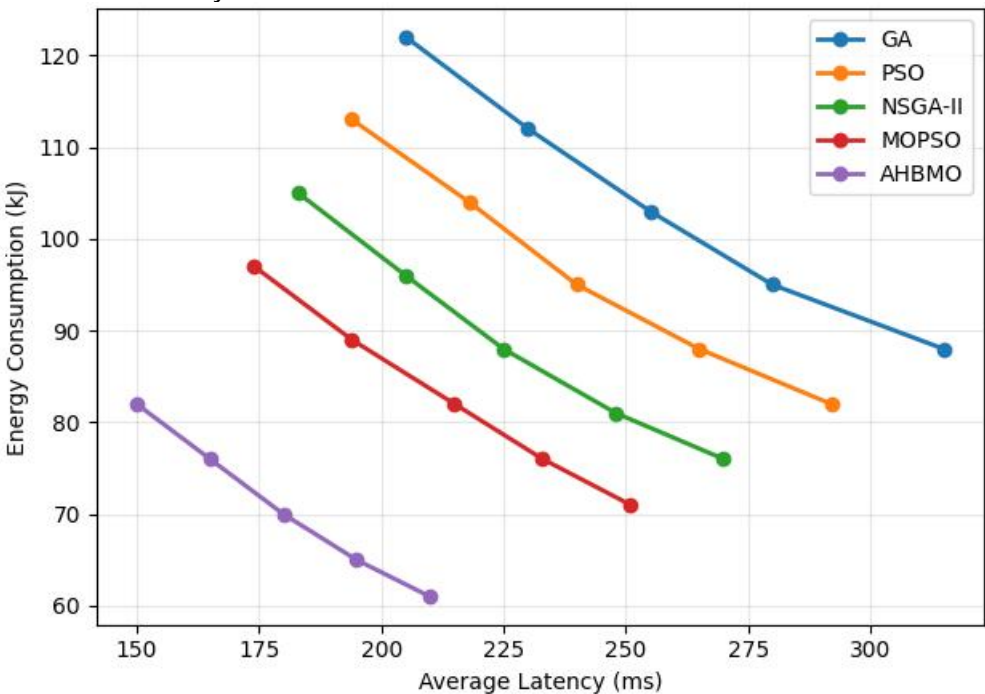


Figure 8. pareto-front comparison

Figure 8 illustrates the different Pareto fronts produced by five algorithms, where latency is shown on the horizontal axis while energy consumption is shown on the

vertical axis. The expected Pareto front produced using AHBMO should cover the majority of the Pareto fronts that are produced by its competitors. Notably, AHBMO achieves lower latency as well as lower energy consumption at the same time. This behavior shows that AHBMO does not merely minimize one of the objectives while compromising on the other objectives.

For instance, one of the AHBMO solutions can achieve latency of 155 ms and energy consumption of 81 kJ while another may achieve 205 ms latency and energy consumption of 62 kJ. The first solution may be used in applications where latency is of utmost importance, while the second may be used in cases where energy consumption is of utmost importance.

5.5.10 Hypervolume Comparison

Hypervolume is a metric to measure the quality of resulted Pareto fronts. Higher hypervolume values signify better convergence and diversity. The AHBMO demonstrates hypervolume of 0.83 in comparison to 0.61 with GA, proving a marked amelioration in the convergence of the Pareto front as well as diversity. Such a significant hypervolume occurs because of the adaptive search method and the memory of non-dominated solutions present in the Pareto archive.

5.5.11 Overall Comparative Analysis

In general, the results obtained from this investigation show that AHBMO provides the best performance based on any of the metrics used. In comparison with the GA method, it gives a 39.9% decrease in the latency period, 35.3% decrease in energy consumption, 29.7% decrease in costs of the cloud service, and 56.5% decrease in SLA violations. Resource utilization and throughput have increased by 18.9% and 21.1%, respectively.

The described improvements should be understood as the effect due to the integrated architecture rather than only due to the optimization approach used in this research. The overall result occurs as a result of interaction of four significant mechanisms, which are:

- AI workload prediction that makes resource provisioning proactive.
- QoS-aware task classification that helps to prioritize clinically significant workloads.
- Adaptive transmission in IoT that helps to reduce unnecessary communication energy consumed.
- Task placement based on AHBMO that aims at finding a balance in the resource allocation on edge-cloud infrastructure level.

Thus, the suggested approach is able to solve both computational and communication problems related to management of telemedicine resources and services. The conclusion

based on the received results is that the combination of IoT, edge computing, cloud computing, AI prediction, and multi-objective optimization can provide significant advantages over traditional approaches to the resource allocation process.

The increase in SLA compliance is extremely important because when it comes to healthcare applications, obtaining energy efficiency or cost-effectiveness should not compromise important patient services. The QoS-supported optimization mechanism prioritizes the execution of emergency tasks, ensuring the utilization of appropriate edge resources. The importance of diminishing energy consumption is emphasized when dealing with IoT telemedicine solutions. The design of many wearable and portable medical sensors implies having a limited capacity of batteries. Therefore, relying on constant transmission of unnecessary data would shorten their lifespan. The function of adaptive transmission solves the problem by providing prompt transmission of clinically significant information while enabling the aggregation or suppression of other unnecessary data.

The cost reduction associated with using the cloud is a demonstration of the profitability of distributing workload across edge and cloud resources. In the case of using a cloud-only architecture, a significant part of the healthcare workloads would be assigned to the centralized infrastructure. The framework proposed in the current paper assesses if a particular workload can be processed locally or at the edge before applying cloud resources. The effectiveness of resource utilization indicates that the method developed is capable of better allocation of workloads. Unlike some nodes being overloaded and some others being underutilized, AHBMO chooses the best resource configuration to achieve equity of task distribution.

The results of convergence indicate that the adaptive search algorithm can obtain high-quality solutions faster, which is critical for dynamic environments of different telemedicine systems since the requirements for the resources can change often. At the same time, there is an imbalance between optimization quality and the computational cost of using the optimization algorithm. Highly complex optimizers need extensive computational resources themselves. As a result, further work should aim at creating lightweight implementations of AHBMO that will find the optimal execution interval allowing an ideal ratio between quality and cost.

It is also essential to understand that the obtained results are based on the simulation data rather than real-world performance. Thus, the next step in this research should involve the implementation of the entire framework and substitution of the simulation data with the values gathered through statistical experimental measurements. In general, the results of research confirm the assumptions suggested regarding the

idea of using an adaptive AI-based multi-objective resource management framework in IoT–Edge–Cloud systems to enhance the efficiency and quality of service in telemedicine applications and reduce energy consumption and cloud resource expenses.

6. Conclusion

This article has introduced an adaptable multi-objective IoT-Edge-Cloud resource optimization system driven by AI that is directed towards telemedicine applications while considering their quality of service (QoS) aspects. The system that was proposed combines IoT devices of the medical nature and gateways, enabling computing on the edge and in the cloud along with the use of AI-based workload forecasting, adaptive transmission of data, classification of relevant tasks in this respect and Adaptive Honey Bee for the optimization objectives. Also, a mathematical model has been developed with the aim of minimizing the end-to-end latency together with the cost of cloud services and the amount of violations of the SLAs.

The AHBMO algorithm being offered is dynamically looking for solutions in resource allocation while varying its exploration and exploitation capacity basing on the progress it accomplished and workload variation that it operates with. The penalty system contributes to making sure that critical healthcare tasks receive priority in their computations. The presented framework is a complex approach to resource management in telemedicine.

Future research will have a practical application of the framework in Python language. It will be tested on a large scale with a varying number of patients and nodes. The efficiency of the framework will be measured in terms of the amount of time taken and costs incurred. Further modifications will also aim at learning others about privacy and confidentiality in federated learning.

References

- [1] M. A. Razdan and S. Sharma, "Internet of Medical Things (IoMT): Overview, Emerging Technologies, and Case Studies," 2022.
- [2] A. Al-Masri, "An Edge-Based Resource Allocation Optimization for the Internet of Medical Things (IoMT)," in *Proc. IEEE 3rd Eurasia Conf. Biomedical Engineering, Healthcare and Sustainability (ECBIOS)*, 2021, pp. 1–6.
- [3] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge Computing: Vision and Challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.
- [4] M. Satyanarayanan, "The Emergence of Edge Computing," *Computer*, vol. 50, no. 1, pp. 30–39, 2017.

- [5] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A Survey on Mobile Edge Computing: The Communication Perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.
- [6] D. Saxena and A. K. Singh, "Workload Forecasting and Resource Management Models Based on Machine Learning for Cloud Computing Environments," 2021.
- [7] H. Mohamed, E. Al-Masri, O. Kotevska, and A. Souri, "A Multi-Objective Approach for Optimizing Edge-Based Resource Allocation Using TOPSIS," *Electronics*, vol. 11, no. 18, Art. no. 2888, 2022.
- [8] S. K. Gupta and A. K. Singh, "Internet of Medical Things (IoMT): Applications and Challenges in Healthcare," 2020.
- [9] M. A. Al-Masri, "Internet of Medical Things and Edge Computing for Improving Healthcare in Smart Cities," *Mathematical Problems in Engineering*, vol. 2022, Art. no. 5776954, 2022.
- [10] S. Rahman, M. A. Hossain, and M. S. Hossain, "Internet of Medical Things: A Review of Applications, Technologies, and Challenges," 2021.
- [11] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A Survey on Mobile Edge Computing: The Communication Perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.
- [12] A. A. Abdellatif, A. Mohamed, C. F. Chiasserini, M. Tlili, and A. Erbad, "Edge Computing for Smart Health: Context-Aware Approaches, Opportunities, and Challenges," 2020.
- [13] J. Fang, S. Lu, et al., "An Efficient Computation Offloading Strategy with Mobile Edge Computing for IoT," *Micromachines*, vol. 12, no. 2, Art. no. 204, 2021.
- [14] A. Chandy, "Smart Resource Usage Prediction Using Cloud Computing for Massive Data Processing Systems," *Journal of Information Technology and Digital World*, vol. 1, no. 2, pp. 108–118, 2019.
- [15] H. Mohamed, E. Al-Masri, O. Kotevska, and A. Souri, "A Multi-Objective Approach for Optimizing Edge-Based Resource Allocation Using TOPSIS," *Electronics*, vol. 11, no. 18, Art. no. 2888, 2022.
- [16] D. Saxena and A. K. Singh, "Workload Forecasting and Resource Management Models Based on Machine Learning for Cloud Computing Environments," 2021.
- [17] N. Sethi, S. Singh, and G. Singh, "Multiobjective Artificial Bee Colony Based Job Scheduling for Cloud Computing Environment," *International Journal of Mathematical Sciences and Computing*, vol. 4, no. 1, pp. 41–55, 2018.
- [18] X. Wu, J. Gan, S. Chen, X. Zhao, and Y. Wu, "Optimization Strategy of Task Offloading with Wireless and Computing Resource Management in Mobile Edge

Computing," *Wireless Communications and Mobile Computing*, vol. 2021, Art. no. 8288836, 2021.

[19] E. Al-Masri, "An Edge-Based Resource Allocation Optimization for the Internet of Medical Things (IoMT)," in *Proc. IEEE 3rd Eurasia Conf. Biomedical Engineering, Healthcare and Sustainability (ECBIOS)*, 2021, pp. 1–6.