Online-Journals.org

iJOE International Journal of Online and Biomedical Engineering

iJOE | eISSN: 2626-8493 | Vol. 22 No. 6 (2026) | OPEN ACCESS

https://doi.org/10.3991/ijoe.v22i06.61555

SPECIAL FOCUS PAPER

Intelligent Hyper-Heuristic Algorithm for Optimizing Application Placement in Fog-Healthcare

Ankur J. Goswami[1] (✉), Asokan Vasudevan[2], Anil Managutti[1], Sanjeev Punia[1], Geetha Subramaniam[2]

[1]Sankalchand Patel University, Visnagar, Gujarat, India

[2]INTI International University, Nilai, Malaysia

ajgoswamice_spce@spu.ac.in

ABSTRACT

Fog computing has become essential for solving latency-constrained applications, especially for instances of healthcare monitoring. However, efficient placement of the applications on the fog network still remains a problem due to factors such as resource constraints, as well as fluctuating workload. The traditional models have limitations in terms of the effective utilization of resources and minimizing latency. To overcome such limitations and to have a faster processing rate, this work presents a hyper-heuristic optimization-based algorithm for healthcare fog computing. For the development of an ECG (electrocardiogram) application, latency is considered a primary optimization target while meeting the resource constraints of the Fog nodes. The proposed hyper-heuristic algorithm can choose simple low-level heuristics based on the network conditions and workloads. Simulations are conducted, and the proposed model is compared against the default FCFS (First-Come, First-Served) policy and state-of-the-art algorithms in terms of delay, fog utilization, application admission rates, energy consumption, and total cost. The results clearly show that the proposed hyper-heuristic outperforms existing methods by up to 70% in terms of fog utilization and close to ~98% admission rates with significantly lowered delay. The outcomes show that using a hyper-heuristic for application placement is effective, especially for real-time healthcare applications such as ECG monitoring.

KEYWORDS

fog computing, hyper-heuristic algorithms, application placement, latency optimization, electrocardiogram (ECG), event prediction, healthcare systems

1 INTRODUCTION

The technological progression over the years has led to changes in practices in the healthcare sector and new approaches to patient care as well as the manner of storing and retrieving medical data [1], [2]. One such technological shift is fog computing, which brings the services provided by cloud computing closer to the network periphery. Fog computing deploys computation, storage, and

Goswami, A. J., Vasudevan, A., Managutti, A., Punia, S., Subramaniam, G. (2026). Intelligent Hyper-Heuristic Algorithm for Optimizing Application Placement in Fog-Healthcare. *International Journal of Online and Biomedical Engineering (iJOE)*, 22(6), pp. 54–71. https://doi.org/10.3991/ijoe.v22i06.61555

Article submitted 2026-03-15. Revision uploaded 2026-04-03. Final acceptance 2026-04-07.

networking facilities closer to data origin and consumers so that it can minimize latency in data processing activity [3]. Another advantage of physical closeness is that it becomes easy to make a quick analysis of data, especially if the healthcare environment is involved. Application of fog computing in healthcare enables the real-time analysis of patients' data, hence allowing faster diagnosis and treatment [4], [5]. This is because processing is done at the edge, and hence, less data is sent to the main servers. Centralized data processing reduces the transmission of health information over computer networks. IoT consumers are enabled by fog computing without burdening central systems as the number of IoT devices increases.

Fog computing [6] defines application placement as the thoughtful assignment of computational tasks among a variety of nodes in the network, including edge devices, fog nodes, and cloud servers. Effective placement of applications is necessary for maximizing system performance, efficient resource utilization, and ensuring Quality of Service (QoS) prerequisites are fulfilled [7]. In health applications where timeliness and data volume are important, the necessity of optimal application placement is undeniable. Figure 1 details the framework of a fog and cloud-integrated healthcare system, which collects instant data from patients in multiple locations, performs local processing through gateways, and executes analysis in cloud servers with the aid of artificial intelligence (AI) software. Every healthcare situation uses multiple communication protocols and devices to send patient information for rapid diagnosis and intervention, resulting in a highly linked and expandable healthcare infrastructure. Figure 1 displays the need for application placement to conquer various clinical monitoring tasks spread across multiple network layers, using cloud servers for processing of relevant data.

As shown in Figure 1, the monitoring devices (such as wearable sensors, smartwatches, and medical devices) capture real-time health data from patients, such as ECG signals, speech data, vital signs, or EEG (Electroencephalogram) readings. These devices communicate with nearby local gateways using technologies such as Bluetooth 4.0 or WLAN/GPRS (Wireless Local Area Network/General Packet Radio Service), establishing a network in mobile, home, hospital, and emergency settings. In a hospital setting, a smart shirt with multiple sensors connects to gateways for vital sign monitoring. The LAN (Local Area Network) uses technologies such as 802.11, 802.15.4, WLAN, and GPRS to communicate with the cloud and perform local processing or data aggregation before sending application requests to the cloud for more intensive AI analysis.

Placing applications closer to data sources minimizes the time taken for data to be processed and responses to be generated. Maximizing the task distribution optimally prevents an overloading of specific nodes while also underutilizing others. Adaptive placement techniques accommodate alterations in network conditions and the demands of the workload [9]. Application distribution across numerous nodes improves both fault tolerance and the system's robustness. For example, in a hospital environment filled with devices constantly producing data, a smart application placement method assures.

Overseeing these finite resources while sustaining superior performance is an important difficulty. The increase in both patients and monitoring devices necessitates that the system maintains its performance through scaling. Handling increasing workloads calls for adaptive algorithms that are efficient.

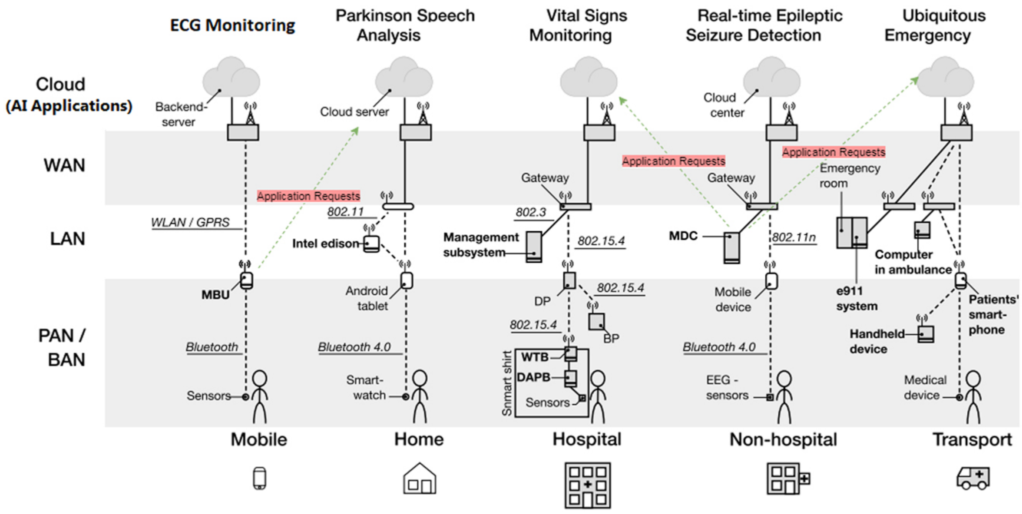

Fig. 1. A multi-tier fog computing architecture used for healthcare applications

Source: Vyan, Desai, and Ruparelia [8].

Improving on these challenges requires the production of complex algorithms that can dynamically optimize application positioning while taking into account the special limitations of fog computing environments. The hyper-heuristic [13], [14] algorithm presents an exciting solution by allowing a comprehensive approach to heuristic selection or generation that can respond to different problem instances and changing scenarios. Based on the severity of ECG event prediction and the difficulty of the fog computing environment, there exists a strong motivation to have powerful application placement strategies to minimize latency and maximize resource usage. It is through the application of Hyper-heuristic algorithms that the research will be able to come up with a robust solution that will address the challenges mentioned above, thus enhancing the performance and reliability of fog-based healthcare systems. This study aims to increase the effectiveness of fog computing in the healthcare context by optimizing the location of ECG event prediction activities. The goal is to use Hyper-heuristic algorithms to realize latency reductions and better use of resources, which is significant in preparing medical responses. Python, together with iFogSim [15], provides a good simulation platform to test and confirm our proposed process. The main contributions of the work are:

- The work develops a novel Hyper-heuristic optimization-based algorithm placement of ECG event prediction applications to healthcare fog computing environments utilizing iFogSim.
- The Hyper-heuristic focuses on minimizing overall latency as the primary objective with resource constraints while placing the applications.
- The system is developed using a hybrid mode of Python with iFogSim, where Python is used for the development of the Hyper-heuristics algorithm responsible for the application placement, and iFogSim is utilized to build the Fog network.
- The effectiveness of the Hyper-heuristic application placement algorithm is evaluated with the default iFogSim placement policy FCFS and state-of-the-art methods using evaluation of fog utilization, delay, and admission rate, energy consumption, and total cost of execution of applications.

2 LITERATURE REVIEW

New studies are investigating numerous techniques designed to overcome this challenge, with attention increasing on hyper-heuristic algorithms because of their adaptability and high efficiency. Strategically locating computational tasks among fog nodes is part of application placement, which strives to achieve performance goals, including the reduction of latency and the improvement of resource utilization. The existing application placement strategies frequently utilize metaheuristic algorithms such as Genetic Algorithms (GAs) [16], Ant Colony Optimization (ACOs) [17], and Particle Swarm Optimization (PSOs) [18], [19]. Though these approaches are capable of delivering nearly ideal results, they may have a hard time being flexible in dynamic fog environments where network conditions and workloads change quickly [20], [21]. The strategic placement of computational assignments across fog nodes to reach performance goals, including minimizing latency and optimizing resource use, is known as application placement. Although these techniques can yield near-ideal outcomes, they could have trouble with flexibility in dynamic fog settings where the network environment and workloads change swiftly [22], [23]. Various recent studies suggest the positioning strategies of contextually aware nodes that consider network bandwidth, node processing capacities, and energy consumption [24]. As an example, in [25], the authors applied the conventional optimization algorithms, i.e., GA, PSO, etc., to solve five weighted multi-objective optimization problems, i.e., FSPGSA (Flow Shop Process Genetic Simulated Annealing) (a combination of GA and simulated annealing) and FPSO (Floating Production, Storage, and Offloading). Nevertheless, these methods were not able to fully address the complexities of real-time uses in healthcare, requiring rapid adjustments to changing conditions as well as strong adherence to latency.

While a number of studies have looked at deploying ECG applications within fog [26], [27] and edge computing architectures, many tend to use rigid placement methods or neglect latency as the principal optimization target. This deficiency reveals the requirement for improved algorithms that can dynamically adapt application deployment for the sake of minimizing latency. Recently, hyper-heuristic [13], [14], [28] algorithms have been applied to resource allocation and task scheduling within computing settings. The most significant success of the current hyper-heuristic algorithms has been the capability to adequately balance exploration and exploitation in the optimization process, which allows avoiding local optima and enhancing the quality of solutions in the long run [29]. In the case of fog computing, such flexibility becomes critical because the optimal layout of the applications can be changed frequently depending on the changing network conditions and workloads. Nevertheless, these techniques might not be adequate on their own. Hyper-heuristic algorithms can enhance optimization of latency by dynamically selecting the most skilled low-level heuristics based on the present condition of the system and thus produce a more sustainable resolution.

The implementation of applications in healthcare in an efficient manner in the fog computing environment is a complex problem to tackle [30]. A promising alternative that might replace hyper-heuristic algorithms is the flexibility and ability to solve problems on a high level. The reduction of the latency as the main parameter of minimization and adjustment to the dynamic nature of the fog environment may be considered to improve the work of such important applications as ECG event prediction [31–33]. Research should continue to create highly specialized hyper-heuristic formats that can be used to support the specialized needs of healthcare systems.

3 MATERIALS AND METHODS

In a fog computing landscape, Fog Application Placement Problem (FAPP) refers to the challenge of arranging the best positioning for computational tasks or services derived from applications across fog nodes, or fog cells. Because fog nodes possess constrained resources (such as processing power, memory, and bandwidth), along with task-specific computational demands, the objective is to improve performance metrics (e.g., latency, energy consumption, cost) while guaranteeing fog node resources are not overwhelmed. The situation gets more complicated thanks to the shifting dynamics of fog networks, in which network conditions and workloads can change over time. Officially, the challenge can be interpreted as an optimization task, which requires the placement of each application service on multiple fog nodes so as to optimize a defined cost metric (such as latency) while respecting resource and workload constraints. The solution seeks to achieve an optimal trade-off between application performance and resource utilization in the fog layer.

An illustration of the FAPP can be depicted in Figure 2, representing the relationships between application services, fog cells, and the objective of optimal placement. The goal of the problem is to map various application services to fog nodes (also called fog cells) to minimize certain objectives such as latency, cost, or energy consumption.

Assuming $A = \{A_1, A_2, A_3, \ldots, A_n\}$ represent the set of application services, where each A_i composed of a set of tasks a_j. For instance, $A_1 = \{a_1, a_2, a_3, a_4, a_5, a_6\}$, $A_2 = \{a_1, a_2, a_3, a_4\}$, and $A_3 = \{a_1, a_2, a_3, a_4, a_5\}$, as shown in Figure 2. Also, $F = \{f_1, f_2, f_3, \ldots, f_m\}$ represents the set of fog cells, where each fog cell is a computational node with certain resource capacities such as CPU, memory, bandwidth, etc. Given W_{ij} represents the workload of the application task from application A_i, which needs to be placed on one or more fog cells.

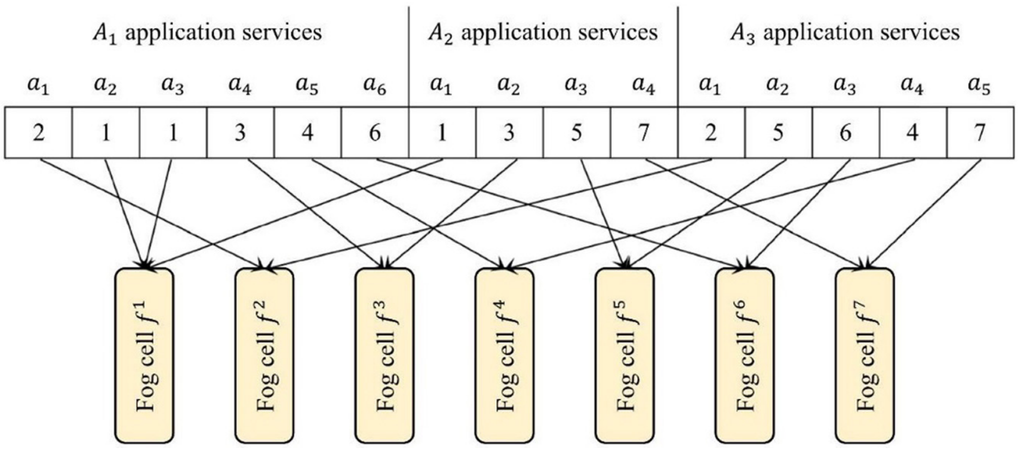

Fig. 2. Application service

Source: Compiled by authors.

The numbers in the matrix under each application service A_1, A_2, A_3 represent the workloads for each task a_j in that application. For example, in A_1, a_1 has a workload of 2, a_2 has a workload of 1, and so on. The objective of the FAPP can be defined as the process of mapping each task $a_j \in A_i$ to one or more fog cells $f_k \in F$ such that certain constraints are satisfied, and a defined cost function is minimized.

3.1 Resource constraints

As each fog cell has limited computational capacity (e.g., CPU, memory), the total workload assigned to a fog cell should not exceed its available resources. Let C_k be the capacity of fog cell f_k. Then, for each fog cell, the sum of workloads assigned to it must satisfy

$$\sum_{a_j \in A} W_{ij} x_{jk} \le C_k$$

Given $x_{jk} = 1$ if that task a_j is assigned to fog cell f_k and $x_{jk} = 0$ otherwise.

The objective function (X) selected for the hyper-heuristic algorithm is to minimize the total latency incurred due to task placement. Assuming L_{jk} represent the latency incurred by assigning task a_j to fog cell f_k. The objective is to minimize the total latency is given by

$$f(X) = \text{Minimize} \sum_{a_j \in A} \sum_{f_k \in F} L_{jk} x_{jk}$$

Subject to resource and placement constraints

$$\sum_{a_j \in A} W_{ij} x_{jk} \le C_k \forall f_k \in F$$

$$\sum_{f_k \in F} x_{jk} = 1 \forall a_j \in A$$

3.2 Hyper-heuristics

Hyper-heuristic algorithms are a class of optimization techniques that operate at a higher level of abstraction compared to traditional heuristics. Instead of solving a problem directly, hyper-heuristics work by selecting or generating low-level heuristics (rules or procedures) to address a given problem. The goal of hyper-heuristics is to automatically choose the best heuristic or combination of heuristics for a specific problem instance, based on the problem's characteristics and current system state. Hyper-heuristics can dynamically switch between different low-level heuristics based on the problem environment or system conditions. This feature is important if heuristic choices are made for working in stochastic and non-stationary contexts, since an optimal heuristic option may vary in the course of time. Thus, hyper-heuristics differ from existing algorithms, as these algorithms are much more universal in their application. This makes them generalizable—usable and portable across various contexts with little adaptation work needed. Hyper-heuristics are designed to choose proper heuristics; therefore, they do not require human initiative or knowledge to tune problem-specific solutions. The fog computing environments are inherently dynamic, the workloads are variable, the networks are unstable, and the fog nodes are resource-constrained.

Through the adaptive choice of effective low-level heuristics, hyper-heuristic algorithms address the challenging, multi-objective optimization challenges. The resources of fog nodes are limited, and efficient task allocation is very important. Hyper-heuristics can balance the processing burden across a number of fog nodes while ensuring adherence to resource constraints, prompting superior comprehensive system performance. As fog networks acquire more devices and applications, the challenging nature of the placement problem increases exponentially. The efficient navigation of extensive solution spaces enabled by hyper-heuristics makes them well-suited to dealing with major fog networks and abundant workloads.

3.3 Hyper-heuristic algorithm

The proposed hyper-heuristic used in this work is designed for solving the FAPP task, particularly for placing the ECG application in a fog computing environment. The goal of the proposed hyper-heuristic algorithm is to achieve an optimal solution through heuristic selection and iterative improvement based on different mappings of solutions in the domain space. The algorithm aims to minimize latency as the primary objective while satisfying resource constraints for each fog node. It uses a pool of low-level heuristics, selects the best heuristic dynamically, and applies it iteratively to improve the candidate solution. The block diagram for the proposed work is shown in Figure 3.

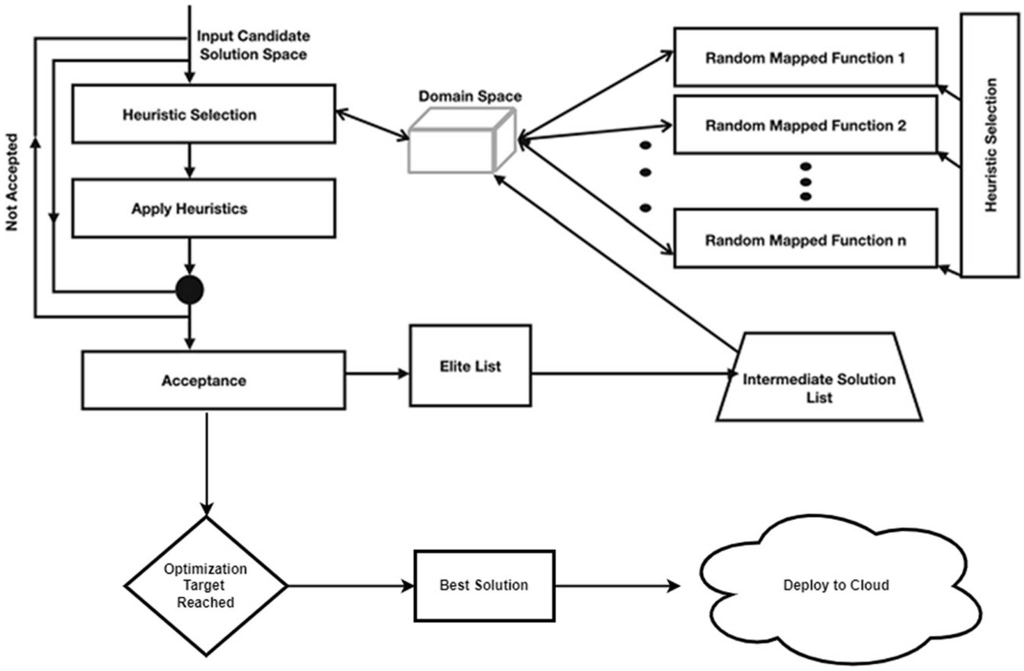

Fig. 3. Architecture of the proposed hyper-heuristic algorithm for ECG FAPP

Source: Compiled by authors.

This represents the initial set of possible placement solutions for the application (in this case, ECG monitoring). These solutions define how application services can be mapped onto available fog resources (fog nodes). The input candidate solution space is the set of all potential placements of tasks a_j on fog nodes f_k.

$$X = \{x_{jk}\}, x_{jk} = \begin{cases} 1, & \text{if task } a_j \text{ is assigned to fog node } f_k \\ 0, & \text{otherwise} \end{cases}$$

Each candidate solution is represented as a matrix X, where a set of low-level heuristics. The hyper-heuristic algorithm selects one heuristic h_i from H based on the current state of the candidate solution. The heuristic selection process is dynamic and adaptive, meaning it depends on the current placement configuration. The heuristic selection process at iteration t can be denoted as

$$h^{(t)} = Select(H, X^{(t)})$$

where $X^{(t)}$ represents the candidate solution at iteration t. Once a heuristic $h^{(t)}$ is selected, it is applied to modify the current solution $X^{(t)}$ to generate a new candidate solution $X^{(t+1)}$. The heuristic modifies the task-to-fog-node mappings by adjusting the values of x_{jk}, which can be represented as:

$$X^{(t+1)} = h^{(t)}(X^{(t)})$$

The heuristic aims to improve the solution by moving towards optimal task placement in terms of minimizing latency or improving resource utilization. The domain space in the algorithm is the complete set of possible candidate solutions. The domain space D consists of all possible task-to-fog-node assignment matrices X, such that

$$D = \left\{ X \mid x_{jk} \in \{0,1\}, \sum_{k=1}^{m} x_{jk} = 1 \forall j \right\}$$

To introduce variability and prevent the algorithm from getting stuck in local optima, randomly mapped functions are applied. These functions generate random solutions in the domain space. The intermediate solutions generated after applying heuristics or random-mapped functions are stored in an intermediate solution list S. This list contains potential solutions that may or may not improve the overall optimization goal. The proposed Hyper-Heuristic Algorithm for ECG Application Placement is as follows:

3.4 Algorithm: hyper-heuristic algorithm for ECG application placement

Inputs:

- Set of ECG application tasks $A = \{A_1, A_2, \ldots, A_n\}$ where $A_i = \{a_1, a_2, \ldots, a_j\}$
- Set of fog nodes $F = \{f_1, f_2, \ldots, f_m\}$ with capacities C_k for each fog node f_k
- Task workloads W_{ij} for each task a_j
- Set of heuristics $H = \{h_1, h_2, \ldots, h_p\}$
- Random Initial candidate solution space $X^{(0)}$

Outputs:
Best task-to-fog node mapping X^((best))
Minimized latency L_jk

Step 1: Initialize

- **a)** Set the initial solution $X^{(0)}$
- **b)** Initialize the elite list $E = \varnothing$.
- **c)** Define the objective function to minimize the latency

$$f(X) = \sum_{a_j \in A} \sum_{f_k \in F} L_{jk} x_{jk}$$

- **d)** Set iteration $t = 0$ and maximum iterations T_{max}

Step 2: Heuristic Selection, at each iteration t, select a heuristic $h^{(t)} \in H$ based on the current candidate solution $X^{(t)}$

$$h^{(t)} = Select(H, X^{(t)})$$

Step 3: Apply Heuristic, Apply the selected heuristic $h^{(t)}$ to the current solution $X^{(t)}$, generating a new candidate solution $X^{(t+1)}$

$$X^{(t+1)} = h^{(t)}(X^{(t)})$$

Step 4: Evaluate Resource Constraints, for each fog node, ensure that the total workload does not exceed the node's capacity C_k. If a fog node is overloaded, reject the solution

$$\sum_{a_j \in A} W_{ij} x_{jk} \le C_k \forall f_k \in F$$

If constraints are violated, return to Step 2 with a new heuristic selection.

Step 5: Acceptance Criteria, Calculate the objective function $f(X^{(t+1)})$ representing the total latency

$$f(X^{(t+1)}) = \sum_{a_j \in A} \sum_{f_k \in F} L_{jk} x_{jk}$$

Step 6: Update elites, If the new solution $X^{(t+1)}$ improves the latency i.e. $f(X^{(t+1)}) < f(X^{(t)})$, accept the solution and update the elite list E.

$$E = E \cup \{X^{(t+1)}\} \text{ if } f(X^{(t+1)}) < f(X^{(best)})$$

Step 7: Update solution after each iteration, update the best solution

$$X^{(best)} = \arg \min_{X \in E} f(X)$$

Step 8: Termination, if the optimization target is reached or the maximum number of iterations T_{max} is exceeded, terminate.

Step 9: Output the best solution $X^{(best)}$ representing the optimal mapping of ECG tasks to fog nodes, along with the minimized total latency.

The elite list E contains the top-performing solutions. These are the solutions that have been accepted based on their performance in optimizing the objective function (e.g., minimizing latency).

$$E = \{X^{(best_1)}, X^{(best_2)}, \ldots, X^{(best_r)}\}$$

Each solution in E represents one of the top candidates found by the algorithm during previous iterations. At each iteration, the new candidate solution $X^{(t+1)}$ is evaluated and either accepted or rejected based on its performance. Let the objective function be (X) to represent the total latency of task placement. The acceptance criteria can be formulated as

$$X^{(t+1)} \textit{ is accepted if } f(X^{(t+1)}) < f(X^{(best)})$$

Where, $X^{(best)}$ is the best solution found so far. By iterating through the solution space using hyper-heuristics, the system aims to find an optimal or near-optimal solution for the placement of ECG application tasks on fog nodes.

4 RESULTS

The proposed study is designed to mimic a fog computing environment, which is done using the iFogSim toolkit. The network has fog nodes that have varying resource capabilities and constraints that include CPU, memory, and bandwidth. The condition of the experiments is to compare the performance of various application placement algorithms, such as Hyper-Heuristic, FS-PGSA, FS-PSO [25], and default FCFS, which are provided in iFogSim when their workloads increase. Tasks, such as ECG monitoring application ones, are generated dynamically and allocated to fog nodes, with a range of tasks between 100 and 500. The simulations consider such measures as latency, energy usage, cost, and fog usage. These metrics are used to evaluate the performance of the algorithms on various time periods to demonstrate the effectiveness of each strategy in the allocation of resources, delay reduction, and energy and cost efficiency management. Using iFogSim simulations, it ensures network conditions and workload variations that are realistic, allowing us to evaluate the effectiveness of the hyper-heuristic algorithm in comparison with other methods.

Table 1. Experimental parameters used to evaluate the proposed hyper-heuristic FAAP solution

Parameter	Value/Range	Description
Number of Tasks	100–500	The number of tasks generated during the simulation for application placement.
Network Bandwidth	100 Mbps	The available bandwidth between fog nodes and between fog nodes and the cloud.
Maximum Iterations	500	The maximum number of iterations for the hyper-heuristic algorithm.
Cost per Task	0.1–0.5 $/unit	The cost associated with processing each task depends on the fog node resources.
Elite List Size	10	The number of top-performing solutions stored in the elite list during optimization.
Application type	ECG	ECG signals are used where the application is to identify or predict the correct ECG event.

Source: Compiled by authors.

Table 1 shows key parameters used for experiments to evaluate the performance of the hyper-heuristic algorithm. Also, many evaluation metrics are used by the hyper-heuristic algorithm for FAPP for the ECG monitoring application to assess the performance, such as Latency (L_{jk}), Fog utilization (%) U_k and Admission Rate.

Resource utilization is the percentage of total available resources (such as CPU, memory, and bandwidth) used by the fog nodes. Efficient resource utilization ensures that fog nodes are neither under-utilized nor overburdened and is given by

$$U_k = \frac{\sum_{a_j \in A} W_{ij} x_{jk}}{C_k}$$

Admission rate (AR) refers to the percentage of tasks or application services that are successfully placed on fog nodes. This parameter indicates the system's ability to handle incoming workloads. The admission rate is calculated as the ratio of successfully placed tasks to the total number of tasks

$$A = \frac{\text{Number of tasks successfully placed on fog nodes}}{\text{Total number of tasks}}$$

4.1 Latency

Figure 4 displays the latency (measured in milliseconds) over a series of run-time periods (minutes) comparing two variations when applications are placed in an FCFS manner and a Hyper-Heuristic manner.

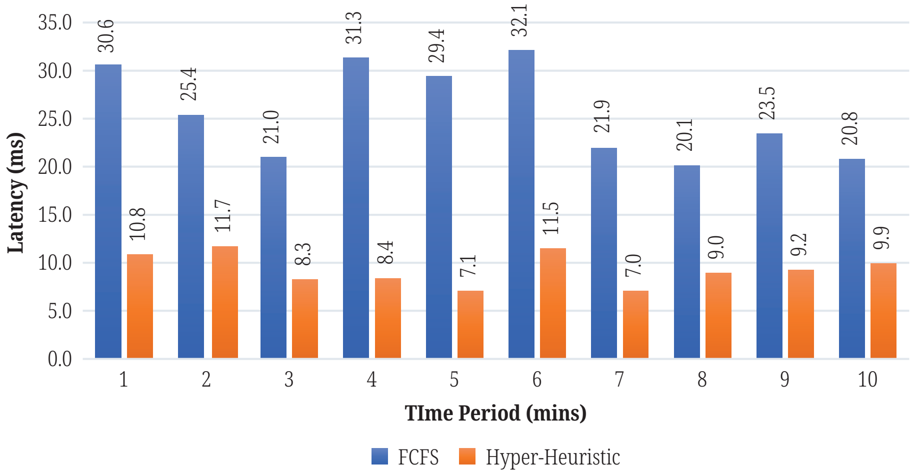

Fig. 4. Observed the latency of the FCFS compared with the proposed hyper-heuristic

Source: Compiled by authors.

The results clearly show that using the hyper-heuristic algorithm leads to a substantial reduction in latency across all time periods. This latency reduction is essential to latency-sensitive applications such as ECG monitoring, where real-time processing of data and timely responses are paramount to effective healthcare monitoring. The delay improvement is also consistent in all periods, which implies that the hyper-heuristic method is strong and can retain its performance over time. In real-time systems like healthcare monitoring systems (e.g., ECG monitoring), the consequent reduction in delays corresponds to faster data analysis and the resulting faster decision-making, which enhances patient outcomes.

4.2 Fog utilization (%)

Figure 5 compares the number of requests processed and fog utilization (FU%) over simulation time (in minutes) between the Hyper-Heuristic algorithm and FCFS. Additionally, the total number of requests in the system over time is also displayed for comparison. The total number of application requests being generated over time remains fairly consistent throughout the simulation, fluctuating around 100% utilization, indicating that the system continues to receive a steady number of requests as the simulation progresses. The fluctuations in this line show the variability in the request load, but overall, the request load remains high.

The Hyper-Heuristic strategy shows a consistent increase in the number of requests processed and fog utilization over time. At the start of the simulation, the number of requests handled by the hyper-heuristic is relatively low but increases steadily, and by the end of the simulation (180 minutes), the hyper-heuristic algorithm handles approximately 85–95% of the total requests.

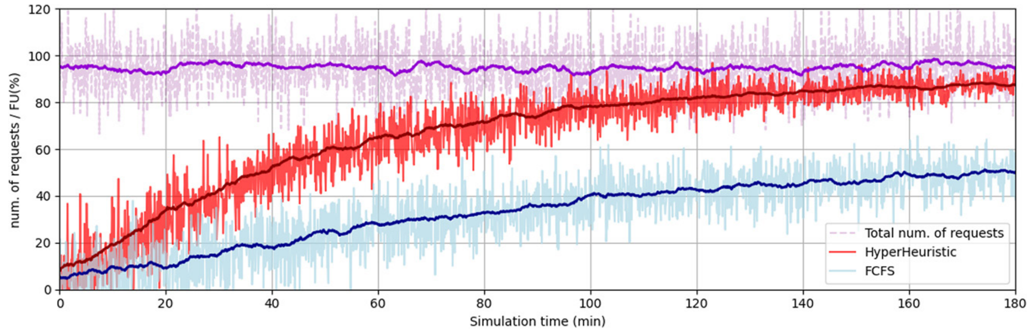

Fig. 5. Number of requests processed and fog utilization (FU%) over simulation time (mins) between FCFS and the hyper-heuristic algorithm

Source: Compiled by authors.

The higher fog utilization in hyper-heuristic implies that the system can handle a greater number of tasks locally on fog nodes, reducing the need to offload tasks to the cloud, which can lower latency and energy consumption. It is evident that hyper-heuristic algorithms are effective in maximizing the utilization of resources in fog computing settings, especially in latency-sensitive ECG monitoring.

4.3 Admission rate

The admission rate, as shown in Figure 6 for the hyper-heuristic algorithm, remains consistently high, between 94% and 99% throughout all time periods, demonstrating that the hyper-heuristic algorithm effectively admits and processes most tasks without overloading the fog nodes. The high admission rate shows the efficiency of the algorithm in assigning resources to tasks by satisfying resource constraints such that the majority of tasks are executed on fog nodes. The hyper-heuristic has a high and constant admission rate, which implies that the hyper-heuristic algorithm is very successful in managing resources, allocating tasks, and enabling the system to cope with nearly any incoming workload.

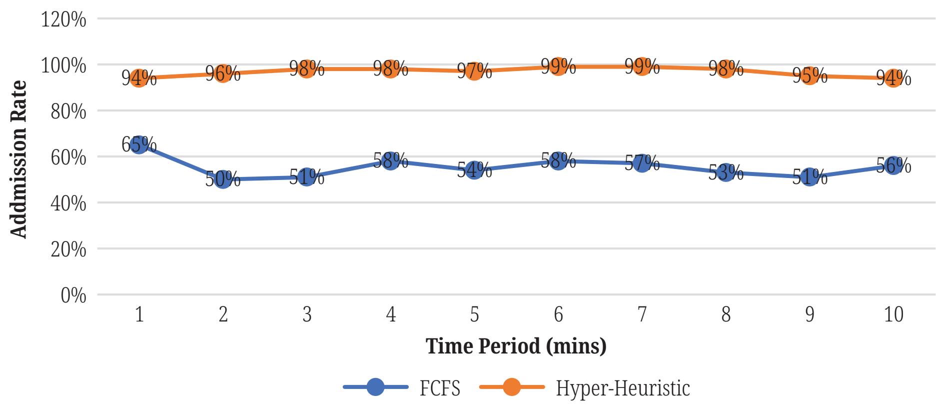

Fig. 6. Admission rate achieved for FCFS and the hyper-heuristic algorithm

Source: Compiled by authors.

The admission rate for FCFS is significantly lower, fluctuating between 50% and 65%, indicating that without an optimized placement strategy, the system is unable to admit and place a large number of tasks.

4.4 Energy consumption

Figure 7 shows the relationship between the number of tasks and energy consumption (kWh) for methods FS-PGSA, FS-PSO, Hyper-Heuristics, and FCFS. As the number of tasks increases, FCFS exhibits the highest energy consumption, rising sharply as the task count reaches 500, indicating inefficient resource use. FS-PSO tends to have a moderate level of energy consumption that rises moderately along with increasing tasks, whereas FS-PGSA tends to be significantly lower yet begins to be on the rise after 400 tasks. The lowest level of energy consumption is always observed in Hyper-Heuristics, which exhibit insignificant increases in energy consumption even when the number of tasks is increased to 500. This implies that hyper-heuristic techniques are very effective in controlling energy consumption as opposed to other techniques and are therefore the best in systems where power is important, e.g., in fog computing systems. The findings indicate that hyper-heuristics are effective in energy consumption optimization as the task load is scaled up.

Figure 8 shows the relationship between the number of tasks and the total cost in dollars ($) for different scenarios. As the number of tasks increases, the FCFS method shows a steep rise in cost, reaching approximately $70 at 500 tasks, highlighting its inefficiency in managing costs. FS-PSO demonstrates moderate cost management, stabilizing around $40, while FS-PGSA maintains a relatively low cost, peaking at $20–$30 as the number of tasks increases.

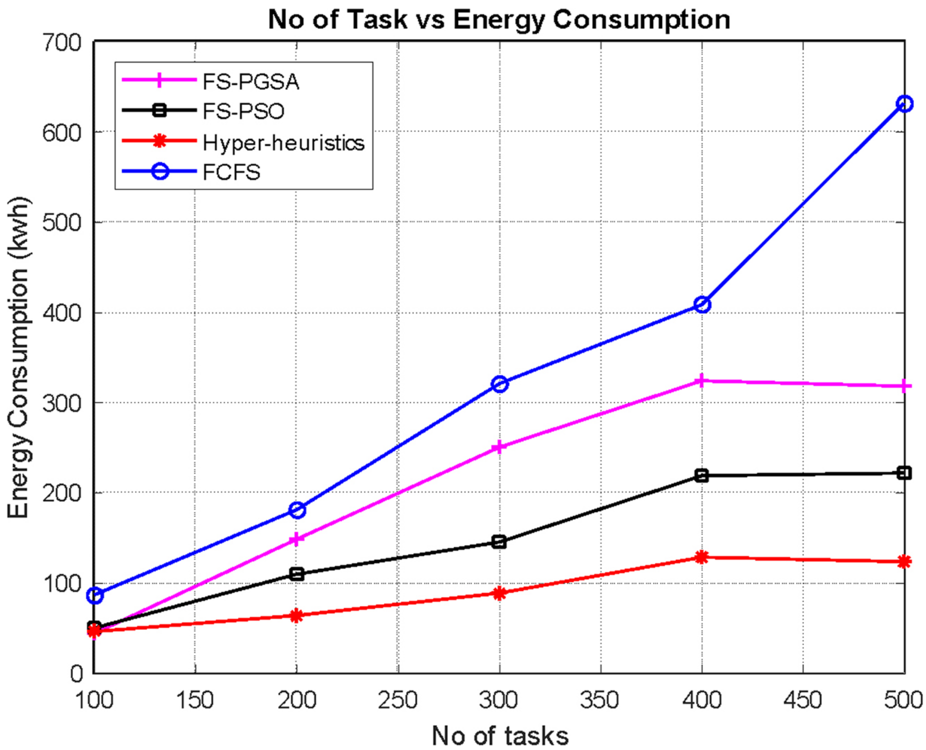

Fig. 7. Comparison of energy consumption (kWh) versus the number of tasks for different application placement strategies

Source: Compiled by authors.

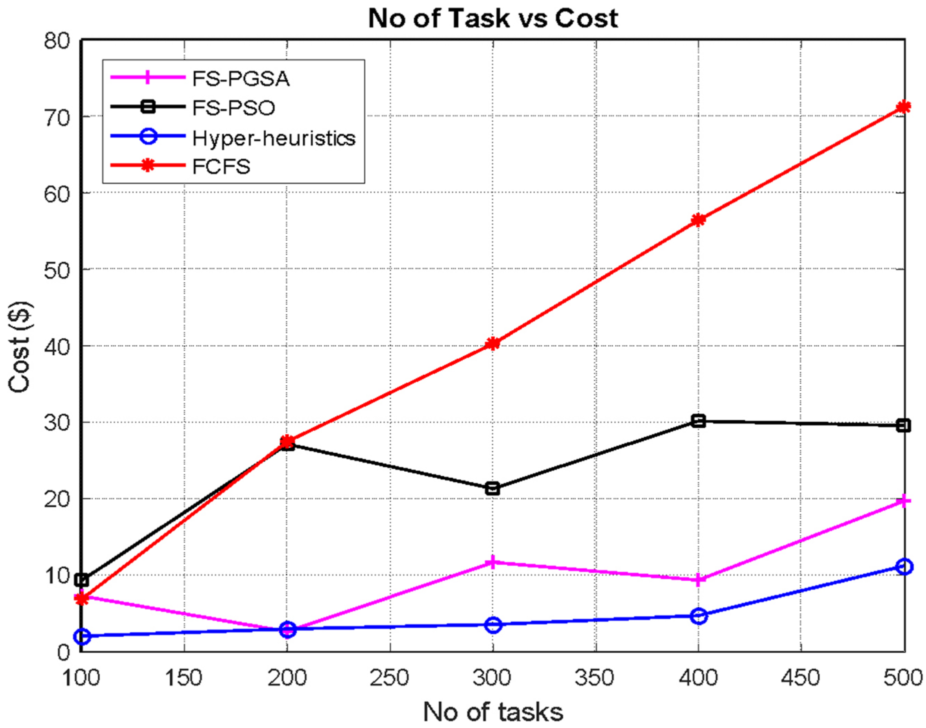

Fig. 8. Comparison of total cost ($) versus the number of tasks for different methods

Source: Compiled by authors.

The Hyper-Heuristics method consistently demonstrates the lowest total cost, remaining well under $15 even at higher task loads. This indicates that the hyper-heuristic approach is significantly more cost-effective in handling larger numbers of tasks, compared to the other methods.

5 CONCLUSION AND FUTURE SCOPE

This work presents a hyper-heuristic optimization-based algorithm for FAAP in healthcare fog. For the development of an ECG application, latency is considered a primary optimization target while meeting the resource constraints of the Fog nodes. The results demonstrate that the proposed hyper-heuristic algorithm significantly outperforms the default FCFS present in iFogSim for FAPP. The hyper-heuristic achieved up to 75% higher fog utilization and consistently lower latency, ensuring more efficient task processing and resource management. The admission rate for tasks remained close to ~98% with the hyper-heuristic, compared to lower rates with FCFS, indicating better system capacity handling. In addition to that, the ability of the hyper-heuristic to be flexible enabled it to manage workloads successfully, which led to improved use of the fog node and reduced delays. The results highlight that the algorithm can be effectively used to handle high and low workloads, and it is an ideal candidate when the application is sensitive to latency, such as real-time healthcare ECG monitoring. Although the algorithm provides promising outcomes, it has some drawbacks; the computational costs of the hyper-heuristic can grow with a larger network and more diverse types of fog nodes and present potential additional overhead in very unpredictable systems. Future studies should focus on enhancing the heuristic selection mechanism to increase scalability and reduce the number of computations required. When there is a necessity to remember about Fog settings, it is also possible to consider the incorporation of other meta-heuristics like ACO and PSO algorithms. Furthermore, the combination of energy efficiency, the usage of fog, and other secondary optimization goals would be favorable to the overall optimization performance.

6 AUTHORS CONTRIBUTIONS

Ankur Goswami: He conceptualized the research problem, designed the overall system architecture, and led the development of the hyper-heuristic optimization framework for healthcare fog computing.

Asokan Vasudevan: He contributed to the formulation of the optimization objectives, algorithmic design, and critical analysis of fog resource constraints and workload dynamics.

Anil Managutti: He was responsible for simulation design, performance evaluation, and comparative analysis with existing state-of-the-art algorithms.

Sanjeev Punia: He contributed to the healthcare application modeling, ECG use-case formulation, and interpretation of results from a real-time healthcare perspective.

Geetha Subramaniam: She assisted in the validation of the methodology, result interpretation, and refinement of the manuscript with emphasis on healthcare applicability and system reliability. All authors contributed to manuscript preparation, review, and approved the final version for publication.

7 REFERENCES

[1] A. V. Dastjerdi, H. Gupta, R. N. Calheiros, S. K. Ghosh, and R. Buyya, "Fog computing: Principles, architectures, and applications," in *Internet of Things*, 2016, pp. 61–75. https://doi.org/10.1016/B978-0-12-805395-9.00004-6

[2] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the Internet of Things," in *Proceedings of the 1st ACM Mobile Cloud Computing Workshop (MCC '12)*, pp. 13–15, 2012. https://doi.org/10.1145/2342509.2342513

[3] R. Das and M. M. Inuwa, "A review on fog computing: Issues, characteristics, challenges, and potential applications," *Telematics and Informatics Reports*, vol. 10, p. 100049, 2023. https://doi.org/10.1016/j.teler.2023.100049

[4] R. Jain, M. Gupta, A. Nayyar, and N. Sharma, "Adoption of fog computing in Healthcare 4.0," in *Fog Computing for Healthcare 4.0: Environments. Technology, Society and Future Implications, Signals and Communication Technology*, S. Tanwar, Ed., Springer, Cham. 2021, pp. 3–36. https://doi.org/10.1007/978-3-030-46197-3_1

[5] A. Kumari, S. Tanwar, S. Tyagi, and N. Kumar, "Fog computing for Healthcare 4.0 environment: Opportunities and challenges," *Computers & Electrical Engineering*, vol. 72, pp. 1–13, 2018. https://doi.org/10.1016/j.compeleceng.2018.08.015

[6] H. K. Apat, R. Nayak, and B. Sahoo, "A comprehensive review on Internet of Things application placement in fog computing environment," *Internet of Things*, vol. 23, p. 100866, 2023. https://doi.org/10.1016/j.iot.2023.100866

[7] M. Goudarzi, H. Wu, M. Palaniswami, and R. Buyya, "An application placement technique for concurrent IoT applications in edge and fog computing environments," *IEEE Transactions on Mobile Computing*, vol. 20, no. 4, pp. 1298–1311, 2020. https://doi.org/10.1109/TMC.2020.2967041

[8] T. Vyas, S. Desai, and A. Ruparelia, "Fog data processing and analytics for health care-based IoT applications," *Fog Data Analytics for IoT Applications*, vol. 76, pp. 445–469, 2020. https://doi.org/10.1007/978-981-15-6044-6_18
[9] A. Hazra, P. Rana, M. Adhikari, and T. Amgoth, "Fog computing for next-generation Internet of Things: Fundamental, state-of-the-art and research challenges," *Computer Science Review*, vol. 48, p. 100549, 2023. https://doi.org/10.1016/j.cosrev.2023.100549
[10] S. N. Srirama, "A decade of research in fog computing: Relevance, challenges, and future directions," *Software: Practice and Experience*, vol. 54, no. 1, pp. 3–23, 2024. https://doi.org/10.1002/spe.3243
[11] H. Djelouat, A. Amira, F. Bensaali, and I. Boukhennoufa, "Secure compressive sensing for ECG monitoring," *Computers & Security*, vol. 88, p. 101649, 2020. https://doi.org/10.1016/j.cose.2019.101649
[12] A. S. Eltrass, M. B. Tayel, and A. I. Ammar, "Automated ECG multi-class classification system based on combining deep learning features with HRV and ECG measures," *Neural Computing and Applications*, vol. 34, no. 11, pp. 8755–8775, 2022. https://doi.org/10.1007/s00521-022-06889-z
[13] E. K. Burke *et al.*, "A classification of hyper-heuristic approaches: Revisited," in *Handbook of Metaheuristics*, vol. 272, 2019, pp. 453–477. https://doi.org/10.1007/978-3-319-91086-4_14
[14] E. Burke *et al.*, "Hyper-heuristics: An emerging direction in modern search technology," *Handbook of Metaheuristics*, vol. 57, pp. 457–474, 2003. https://doi.org/10.1007/0-306-48056-5_16
[15] R. Mahmud, S. Pallewatta, M. Goudarzi, and R. Buyya, "iFogSim2: An extended iFogSim simulator for mobility, clustering, and microservice management in edge and fog computing environments," *arXiv preprint arXiv:2109.05636*, 2021. [Online]. Available: http://arxiv.org/abs/2109.05636 [Accessed: Mar. 6, 2026].
[16] C. Canali and R. Lancellotti, "GASP: Genetic algorithms for service placement in fog computing systems," *Algorithms*, vol. 12, no. 10, p. 201, 2019. https://doi.org/10.3390/a12100201
[17] M. K. Hussein and M. H. Mousa, "Efficient task offloading for IoT-based applications in fog computing using ant colony optimization," *IEEE Access*, vol. 8, pp. 37191–37201, 2020. https://doi.org/10.1109/ACCESS.2020.2975741
[18] W. Abbes *et al.*, "An enhanced binary particle swarm optimization algorithm for service placement in hybrid cloud platforms," *Neural Computing and Applications*, vol. 35, pp. 1343–1361, 2023. https://doi.org/10.1007/s00521-022-07839-5
[19] T. Djemai, P. Stolf, T. Monteil, and J. M. Pierson, "A discrete particle swarm optimization approach for energy-efficient IoT services placement over fog infrastructures," in *Proceedings of 2019 18th International Symposium on Parallel and Distributed Computing (ISPDC)*, 2019, pp. 32–40. https://doi.org/10.1109/ISPDC.2019.00020
[20] M. A. B. Al-Tarawneh, "Bi-objective optimization of application placement in fog computing environments," *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, pp. 445–468, 2022. https://doi.org/10.1007/s12652-021-02910-w
[21] M. Azimzadeh, A. Razaee, S. J. Jassbi, and M. Esnaashari, "Placement of IoT services in fog environment based on complex network features: A genetic-based approach," *Cluster Computing*, vol. 25, pp. 3423–3445, 2022. https://doi.org/10.1007/s10586-022-03571-w
[22] A. H. Abdellatif, A. Singh, A. Aldribi, A. Ortega-Mansilla, and M. Ibrahim, "A novel framework for fog-assisted smart healthcare system with workload optimization," *Computational Intelligence and Neuroscience*, vol. 2022, no. 1, p. 4174805, 2022. https://doi.org/10.1155/2022/4174805
[23] M. Barzegaran, A. Cervin, and P. Pop, "Towards quality-of-control-aware scheduling of industrial applications on fog computing platforms," in *Proceedings of the Workshop on Fog Computing and the IoT*, 2019, pp. 1–5. https://doi.org/10.1145/3313150.3313217

[24] M. Zare, Y. E. Sola, and H. Hasanpour, "Towards distributed and autonomous IoT service placement in fog computing using asynchronous advantage actor-critic algorithm," *Journal of King Saud University – Computer and Information Sciences*, vol. 35, no. 1, pp. 368–381, 2023. https://doi.org/10.1016/j.jksuci.2022.12.006

[25] H. K. Apat, B. Sahoo, V., Goswami, and R. K. Barik, "A hybrid meta-heuristic algorithm for multi-objective IoT service placement in fog computing environments," *Decision Analytics Journal*, vol. 10, p. 100379, 2024. https://doi.org/10.1016/j.dajour.2023.100379

[26] T. N. Gia *et al.*, "Fog computing in healthcare Internet of Things: A case study on ECG feature extraction," in *Proceedings of the IEEE International Conference on Computer and Information Technology*, 2015, pp. 356–363. https://doi.org/10.1109/CIT/IUCC/DASC/PICOM.2015.51

[27] W. Wu, S. Pirbhulal, A. K. Sangaiah, S. C. Mukhopadhyay, and G. Li, "Optimization of signal quality over comfortability of textile electrodes for ECG monitoring in fog computing based medical applications," *Future Generation Computer Systems*, vol. 86, pp. 515–526, 2018. https://doi.org/10.1016/j.future.2018.04.024

[28] M. Maashi, G. Kendall, and E. Özcan, "Choice function based hyper-heuristics for multi-objective optimization," *Applied Soft Computing*, vol. 28, pp. 312–326, 2015. https://doi.org/10.1016/j.asoc.2014.12.012

[29] P. Ryser-Welch and J. F. Miller, "A review of hyper-heuristic frameworks," in *Proceedings of the Evo20 Workshop, AISB*, vol. 2014, 2014. [Online]. Available: https://www.researchgate.net/publication/261361800_A_Review_of_Hyper-Heuristic_Frameworks [Accessed: Mar. 4, 2026].

[30] Naouri *et al.*, "Efficient fog node placement using nature-inspired metaheuristic for IoT applications," *Cluster Computing*, vol. 27, pp. 8225–8241, 2024. https://doi.org/10.1007/s10586-024-04409-3

[31] A. Kumar and P. Banerjee, "Leading with intelligence: How AI is reshaping creative innovation in organizations," *Prabandhan: Indian Journal of Management*, vol. 18, no. 9, pp. 54–62, 2025. https://doi.org/10.17010/pijom/2025/v18i9/174843

[32] H. Sikandar, A. F. Abbas, N. Khan, and M. I. Qureshi, "Digital technologies in healthcare: A systematic review and bibliometric analysis," *International Journal of Online and Biomedical Engineering*, vol. 18, no. 8, pp. 34–48, 2022. https://doi.org/10.3991/ijoe.v18i08.31961

[33] A. U. Adoghe, E. Noma-Osaghae, and R. I. Yabkwa, "Photonic crystal and its application as a biosensor for the early detection of cancerous cells," *International Journal of Online and Biomedical Engineering*, vol. 16, no. 3, pp. 86–94, 2020. https://doi.org/10.3991/ijoe.v16i03.12523

8 AUTHORS

Dr. Ankur J. Goswami is an academic and researcher in the field of computer engineering with research interests in fog and edge computing, Internet of Things (IoT), healthcare computing, and optimization algorithms. His current research focuses on latency-aware resource management and intelligent application placement for real-time healthcare monitoring systems. He is an Associate Professor in the Department of Computer Engineering, Sankalchand Patel University, Visnagar, Gujarat, India (E-mail: ajgoswamice_spce@spu.ac.in, ajgoswami.ce@gmail.com).

Dr. Asokan Vasudevan is a dean and professor at INTI International University, Nilai, Malaysia. His research interests include distributed systems, cloud, fog, and edge computing, intelligent optimization techniques, and scalable healthcare applications. He has contributed to several research works addressing performance optimization and resource efficiency in next-generation computing environments (E-mail: asokan.vasudevan@newinti.edu.my).

Dr. Anil Managutti is with the Dept. of OMFS, NPDCH, Sankalchand Patel University, Visnagar, India. He is a faculty member and researcher with expertise in modeling, simulation, and performance evaluation of distributed and fog computing systems. His research interests include IoT-based systems, resource optimization, and energy-efficient computing. He has actively contributed to comparative analysis and evaluation of state-of-the-art algorithms in fog and edge computing (E-mail: amanagutti.fds@spu.ac.in).

Dr. Sanjeev Punia is a Professor with the Department of Computer Engineering, Sankalchand Patel University, Visnagar, Gujarat, India. His research interests include healthcare informatics, medical signal processing, and the application of advanced computing technologies for real-time clinical monitoring. His contributions emphasize bridging engineering solutions with practical healthcare requirements (E-mail: spce_spce@spu.ac.in).

Dr. Geetha Subramaniam is a faculty member at the Faculty of Health and Life Sciences, INTI International University, Nilai, Malaysia. Her research interests include digital health, healthcare systems engineering, data-driven healthcare solutions, and interdisciplinary applications of computing in medical and life sciences. She is actively involved in research focused on improving the reliability and effectiveness of healthcare technologies (E-mail: geetha.subramaniam@newinti.edu.my).