Journal of Applied Research and Technology

www.jart.icat.unam.mx

ICAT Instituto de Ciencias Aplicadas y Tecnología

Journal of Applied Research and Technology 23 (2025) 564-578

Original

Enhancing Application Placement and Prioritizing Emergency Patient in Medical Cloud-Fog Environments: A Multi-Criteria Decision-Making Approach

B. Arichi[1]* • A.Amraoui[2]

[1]Telecommunication Laboratory Tlemcen (LTT),
Dept. Computer science, Abou Bekr Belkaid University Tlemcen, Algeria
[2]Telecommunication Laboratory Tlemcen (LTT),
Dept. Computer science, Abou Bekr Belkaid University Tlemcen, Algeria

Received: 06 25 2024; Accepted: 03 24 2025
Available: 12 31 2025

Abstract: Fog computing (FC) emerged as a solution to the constraints of cloud computing (CC), particularly in IoT and real-time applications, by providing additional computing resources and services. Its essence lies in processing data at the network's edge via a distributed network of nodes, enabling more efficient, responsive data processing. However, choosing suitable fog nodes that can host and process application modules poses a challenge.

This paper proposes a study to address this challenge by using widely recognized MCDM (multi-criteria decision-making) methods, including AHP (Analytic Hierarchy Process) and TOPSIS (Technique for Order of Preference by Similarity to Ideal Solution). The experiment was conducted using the iFogSim simulator, and the proposed method was evaluated based on CPU power, storage (RAM), latency, and network bandwidth. AHP is used to calculate the priority weights for all evaluation criteria, while the TOPSIS method is used to arrange the appropriate fog nodes. This comprehensive approach aims to facilitate the selection of the most suitable fog nodes for hosting and processing application modules. The simulation results indicate that the proposed method offers notable benefits in placement time, resource gain, energy consumption, and latency in a fog computing environment.

*Corresponding author.
E-mail address: bochra.arichi@univ-tlemcen.dz (B. Arichi).
Peer Review under the responsibility of Universidad Nacional Autónoma de México.

Keywords: **Cloud Computing - Fog Computing –Internet of Things (IoT) - Resource Management- Application Placement – Healthcare - TOPSIS - AHP – iFogSim.**

1. Introduction

As the Internet of Things continues to expand across various fields, it becomes an integral part of our environments and daily activities. The increasing use of IoT also necessitates the efficient processing and storage of the large amounts of data it generates. In cloud-based applications, most data for storage, analysis, and decision-making is sent to cloud data centers (Yousefpour et al., 2019).

Cloud computing is a powerful way to manage and process large amounts of data efficiently. However, when it comes to hosting IoT applications, specific challenges must be addressed, including latency, Network congestion, bandwidth, and security issues. To overcome these limitations, Cisco has introduced a new computing model, fog computing, situated between IoT devices and CC (Ahuja & Deval, 2021).

FC is indeed a distributed computing model that brings cloud capabilities to the edge of the network, where resources are closer to the end-users and data sources. This approach is designed to address certain limitations and constraints associated with traditional cloud computing, especially for latency-sensitive and real-time interaction applications.

Fog computing can enhance many domains. Particularly in healthcare, fog computing has the potential to be extremely useful and relevant through facilitating edge data processing, real-time communication, and decision-making. For instance, fog-based systems can facilitate communication among emergency responders, hospitals, and healthcare providers during emergencies and provide instant access to patient records and real-time data from medical devices.

As a relatively new paradigm, FC introduces several challenges. Among these challenges, due to resource limitations, resource management (RM) issues are among the most important to consider in the fog environment (Varshney et al., 2021). It is important to recognize that RM in fog computing is not just one single problem but comprises multiple challenges across different dimensions and constraints. According to Ghobaei-Arani et al. (2020), RM approaches have been categorized into six primary categories: application placement (AP), resource allocation (RA), resource scheduling (RS), task offloading (TO), load balancing (LB), and resource provisioning (RP). Figure 1 illustrates the various aspects and challenges of resource management. These issues are detailed in [4]. Due to these four issues, RM has gained paramount significance. This is exemplified through the RP, AP, RS, and RA (Varshney et al., 2021).

The first category focuses on application placement, the process of choosing where and how to deploy and execute applications within a fog environment (Ghobaei-Arani et al., 2020). Selecting the fog nodes that will host and run specific applications is necessary to optimize resource usage, latency, performance, and other key metrics.

For fog computing, a variety of algorithms and techniques can be applied to application placement. These algorithms attempt to identify the best location for applications that depend on multiple factors, including network conditions, resource availability, application requirements, and other performance metrics. The primary objective of this study is to enhance comprehension of Fog Application Placement by using Multi-Criteria Decision-Making (MCDM) techniques.

MCDM involves organizing and addressing decision-making and planning challenges that involve multiple criteria or factors. In many cases, there is not a single, definitive optimal solution to these problems. It becomes essential to rely on the decision maker's preferences to distinguish between potential solutions (Majumder, 2015).

The primary focus of this study is the use of a combination of two MCDM methods — specifically, AHP (Analytic Hierarchy Process) and TOPSIS (Technique for Order of Preference by Similarity to Ideal Solution) to address the problem effectively. The paper's following sections are arranged as follows: The related works discussed in Section 2, and the methodology, which includes using MCDM techniques for performance assessment, are detailed in Section 3. Section 4 highlights the Simulation Result and offers a detailed discussion, followed by Section 5, the conclusion.

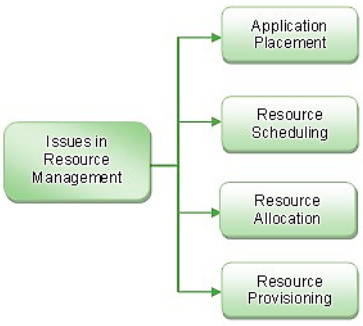


Figure1. Categories of Resource Management Issues (Varshney et al., 2021).

2. Related Works

In this section, we review prior studies on techniques for deploying applications in FC environments. Application placement refers to the decision of where to deploy applications to efficiently utilize resources.

The authors in (Varshney et al., 2021) propose a strategy for managing resources for fog computing environments based on the Analytic Hierarchy Process (AHP) paradigm. This approach addresses significant issues that are required to ensure that intelligent applications perform optimally. They rely on AHP to priorities Fog resources based on quality of experience (QoE) parameters such as average latency, storage, processing time, and network bandwidth. To analyze 99 Fog nodes, the proposed technique determines the ranks and scores of the Fog resources. The Fog resource, which has the lowest rating, will be assigned to the smart application that requires it. The ranking list indicates which resources need to be handled first. Following the initial allocation, they distribute all subsequent Fog resources according to their ranks. This study demonstrates the usefulness of fog monitoring for resource management.

Baranwal & Vidyarthi (2021) introduced an approach to application placement that involves a Fog Orchestrator Node (FON). This orchestrator assesses both the application's requirements and the current state of the fog infrastructure to determine the appropriate fog nodes for application placement. To improve system performance and reliability, the study highlights the advantages of decentralized FON selection and discusses the challenges of centralized fog orchestration. They select FON using the TOPSIS approach, considering specific metrics for Fog Computational Nodes (FCNs) and Fog Gateway Nodes (FGNs). Through simulations, the effectiveness of their method is assessed, demonstrating improvements in system reliability, application location optimization, and resource utilization.

Goudarzi et al. (2020) proposed an approach to application placement consisting of three distinct phases: pre-scheduling, batch application placement, and failure recovery. The first phase of the algorithm aims to provide brokers with a method for orchestrating concurrent workflows involving IoT devices. In the second phase, they present a batch application placement algorithm that utilizes a Memetic Algorithm (MA) based on the genetic algorithms (GA) functions to guide the placement decisions for tasks within each schedule, Finally the Failure Recovery Phase, to address the possibility of task execution failures during runtime, in their system, by keeping records of available servers and regularly reviewing their upcoming task assignments. Based on parameters including failure recovery, maximum iteration number, bandwidth, system size, and decision time analysis, the outcomes indicate the efficacy of the suggested strategy in placing applications in the best possible positions.

Mahmud et al. (2018) proposed a policy for deploying application modules in an orderly manner across distributed fog nodes based on latency sensitivity. The authors introduced two algorithms, with the first one being the module placement algorithm known as "PlaceAppModules" To secure quality of service that meets deadlines within the fog layer. The second algorithm, known as the "ForwardAppModules" strategy for application module forwarding, uses a "first fit" approach to schedule application modules. Additionally, the algorithm includes a relocation phase in which modules are moved from under-occupied nodes to highly occupied nodes as part of the resource optimization process. They evaluated both algorithms, considering factors such as placement time and the percentage of deadlines met. These evaluations were performed in comparison to other latency-aware strategies.

Natesha et al. (2018) highlighted that placing IoT application modules on computationally constrained resources is an NP-hard problem. To address this challenge, they have solved it using a heuristic based on the First-Fit Decreasing strategy. Their proposed method has demonstrated notable improvements, including a substantial reduction in application latency and energy consumption when compared to a benchmark method for Fog-Cloud computing environments.

In (Baranwal et al., 2020), the authors introduced a QoE-aware placement policy for applications that utilizes a Modified-TOPSIS approach to assign ratings to

applications (RoA) and fog computing (RoF) instances, considering various performance metrics. M-TOPSIS is used to prioritize applications based on user expectations and fog instances, taking into account their computational capabilities. M-TOPSIS is a modified version that accounts for the weights of the criteria used to evaluate the alternatives and addresses the rank reversal problem. (The modification is found in the weighted normalized decision matrix). They evaluated the performance of their method in term of " Application placement time", "RG: Resource Gain", "NRR: Network Relaxation Ratio" and "PTRR: Processing Time Reduction Ratio", and the results of this study showcase the benefits of their policy in optimizing application placement, leading to improved performance, reduced network congestion, and enhanced Quality of Experience for IoT applications and their users.

Mahmud et al. (2019) proposed a policy for application placement that is highly sensitive to users' Quality of Experience (QoE). This approach takes into account two critical factors: the user's expectations for the applications and the current state of the Fog computing instances. They have used two distinct fuzzy logic models to streamline the mapping of applications to suitable instances. These models calculate the "Rating of Expectations" for applications and the "Capacity Class Score" for instances. The proposed policy has been confirmed through simulation experiments on "Application placement time", "RG", "NRR", "PTRR", and "QoE gain".

Al-Tarawneh (2022) discusses an application placement algorithm, based on a bi-objective optimization problem that considers both the application's criticality and security requirements. It applies the NSGA-II algorithm to formulate the placement challenge as a bi-objective knapsack problem. The suggested approach aims to determine which Processing Modules, without exceeding each fog node's processing capability, should be placed on each fog node to maximize its security affinity score and overall criticality score. Overall, the outcomes indicate the effectiveness of the suggested technique in optimizing application placement and improving application performance, power efficiency, and security satisfaction rates.

The paper (Mishra et al., 2019) explores the challenges of designing an efficient resource selection and allocation model for fog computing environments, which, by definition, are distributed, scalable, and dynamic. In order to attain an optimal ranking of alternatives in such environments. It proposes an adaptive multi-criteria decision-making (A-MCDM) model. The A-MCDM model outperforms existing MCDM techniques, with overall time complexity O(nm) and O(m) for rating individual alternatives. The A-MCDM technique reduces access time and basic operations by avoiding pairwise comparisons and sorting, which makes it appropriate for adaptive systems. A selection of metrics, including complexity analysis, mean absolute error (MAE), Spearman rank correlation, response time (in seconds), and precision measures, is used to evaluate the A-MCDM model's performance relative to Simplified PROMETHEE-II and PROMETHEE-II. A-MCDM has a lower complexity in both static (O(mn)) and dynamic (O(m)) situations, according to the complexity analysis. The three systems were set up on a MySQL server platform in order to verify the models.

In (Lera et al., 2019), the authors examine the Fog Placement Problem (FPP), which addresses selecting the optimal placement of services in fog computing environments. The study proposed the use of Electre III, an MCDM method, to rank fog node placement alternatives based on multiple performance criteria; including latency, hop count, cost, deployment penalty, and energy consumption. Their study compared Electre III with the weighted average method in a simulated fog computing environment using the Yet Another Fog Simulator (YAFS). In dynamic environments, Electre III proves to be more adaptable than the weighted average method in terms of response time, hop count, energy consumption, cost, and deployment penalty. Through the use of preference, indifference, and veto thresholds, it successfully achieves a balance between several criteria.

Two AHP-based resource allocation strategies for fog-cloud hybrid systems are suggested in this paper (Mishra et al., 2020).In order to minimize delay in the system and enhance resource utilization, these policies optimize task placement by taking into account both compute and network load. In the first method, compute and network resources have predefined weights (eigenvector-based); in the second method, criteria weights are dynamically determined based on real-time system data (SECA-based). Using iFogSim, which divided tasks into dependent subtasks and assigned modules to appropriate nodes using a dependency matrix, the suggested methods were assessed. Task delay, response time, waiting time, compute and network efficiency, and load balancing are important performance measures that are utilized for evaluation. According to experimental data, AHP-based approaches perform more effectively than existing resource allocation strategies, significantly reducing waiting and response times while improving load balancing and resource efficiency.

3. Methodology

Determining the placement of application modules is a significant challenge in fog computing due to both geographical distribution and the constrained computing capacities of fog devices. Therefore, more effective placement strategies are needed to deploy these applications to fog nodes, ensuring that their resource needs are met. Each application module is given a request for resource requirements.

3.1 The system architecture

In Cloud – Fog - IoT environment, a common architectural approach in many studies involves a three-tier: the top tier is the cloud, the middle tier is the fog layer, and the lowest tiers user devices (Ranesh Kumar Naha, Saurabh Garg, Andrew Chan, 2018).

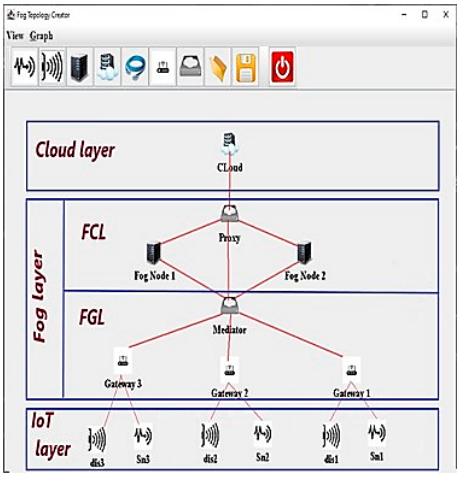


Figure 2. Organizing the Cloud-Fog-IoT Environment

In our study, our attention is directed towards the middle tier, which could have several tiers of fog nodes (Redowan Mahmud, Satish Narayana Srirama, Kotagiri Ramamohanarao, Rajkumar Buyya,, 2019). As a result, we have divided this layer into two sub-layers: Fog Gateway Layers (FGL) positioned at the lower level, proximate to the IoT layer, which are in charge of gathering patient sensor data, and Fog Computational Layers (FCL) situated at the higher level. The processing of the gathered data is the responsibility of this sub-layer as seen in Figure 2.

The components of our system architecture are listed here, along with a description of each level's tasks:

- Cloud Layer (CL): Situated above the Fog layer. It serves as a crucial element of the architecture by offering significant extra resources, these resources are essential when the Fog layer needs more help with data processing and storing.
- Fog Layer (FL): is an intermediate layer located between the IoT and Cloud layers. This layer has several benefits, including enabling real-time data analysis and lightening the cloud's load, but it also has some limitations that need to be addressed. In this study, this layer is consists of : • Fog Computational Layers (FCLs): The higher-tier of the Fog Layer, known as the Fog Computational Layer (FCL), offers computational services and resources to applications for data signal processing, analysis and storing. • Fog Gateway Layers (FGLs): The bottom tier of the Fog Layer, referred to as the Fog Gateway Layer, is positioned in close proximity to the users. Acting as intermediaries between the Fog Computational Layer (FCL) and end-users, they receive data signals and transmit them to the FCL.

Differences in computational intensity and resource capacity exist between Fog Gateway Layer and Fog Computational Layer, with the lower level having fewer resources compared to the upper level. Nevertheless, we assume that each FGL possesses the capability to execute its operations during application placement (specifically, the Patient_Module and Emergency_Module). And the FCL's capabilities will be discuss at a later point.

- IoT Layer: The Internet of Things layer includes different devices like sensors and actuators. These devices have the role of sensing and collecting data from the end-user, and transmitting it to the upper layers for subsequent analysis and processing.

3.2 Application Model

To synchronize with the distributed data flow methodology implemented in the core iFogSim framework, we have structured the Healthcare system application as a Directed Acyclic Graph (DAG), visually depicted in Figure 3. This DAG comprises four distinct application modules, each of which is detailed below:

- Patient_Module: is implemented on gateways, primarily collecting patient data from medical sensors, and then transmitting it to the upper layer.
- Emergency_Module: This module is tasked with computing and identifying the emergency level. This level is used to Rank patients (Gateway nodes) according to service priority, with a higher level patients receiving preferential attention.

The following section discusses this level in detail. Additionally, this module's role includes sending data for processing to the Processing_Module and relaying the results back to the Patient_Module.

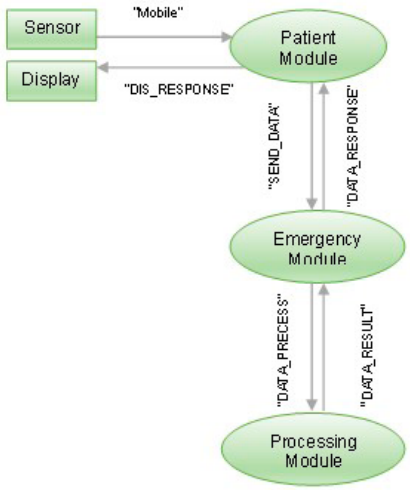


Figure 3. Application model for the Healthcare System.

- Processing_Module: this module enables the processing and storing of data gathered from IoT sensors. The Emergency_Module then receives the findings of these analyses and displays it to the user through the Actuators display.

Given that the Patient_Module (placed on Gateways) and Emergency_Module (placed on the mediator) are preconfigured, our primary focus is on situating the Processing_Module on the Fog node (FCL). To simplify, throughout the rest of the paper, when we mention "application" we are referring to the Application Module that requires placement.

3.3 Application Parameters

Each application module is linked to a request for resource requirements, and each application placement must meet the required application standards. In this scenario, the requirements of each module are depicted in Table 1.

Table 1. Module's characteristics.

Level	CPU(Mips)	Ram(MB)
Patient_Module	10	128
Emergency_Module	10	100
Processing_Module	100	250

3.4 Fog Devices Parameters

A fog device is any component within the network capable of hosting application modules (Gupta et al., 2017). Indeed, in iFogSim, modules are typically assigned to fog devices following a hierarchical structure. This indicates that the iFogSim toolkit prevents interaction between two devices at the same level (Gupta et al., 2017). For example, if a fog device cannot meet a module's requirements, the module is reassigned to an upper-level fog device. Let us consider the available fog infrastructure, which consists of three layers of fog devices: Cloud, FCL, and FGL (Figure 2), each with varying capabilities. The Cloud device has abundant resources, while the Fog layer has limited resources. The fog layer in our scenario comprises four types of fog devices. First, when a mediator requests resources from the Cloud, the proxy server acts as an intermediary. Second, the processing and storing of data collected come under the control of fog nodes. After that, a mediator acts as a liaison for requests from gateways, and each mediator represents a medical service. Lastly, gateways, each representing a patient, are ultimately responsible for collecting data from the medical sensors the patients wear. For testing the proposed work, Table 2 illustrates the resource availability for each fog node.

Table 2. Fog Devices characteristics.

Fog Devices			CPU	RAM	UpBw	DownBw	latency
Cloud layer			44800	40000	100	10000	-
Fog Layer	FCL	Proxy	2800	4000	10000	10000	100
		Fog Node	4000-6000	6000-8000	1000-10000	1000-10000	5-10
	FGL	Med	2800	4000	10000	10000	5
		GW	500	1000	10000	10000	2

Now, the issue is to map the Processing module onto the available Fog resources while considering the modules' constraints specified in Table 1.

3.5. Proposed Work

The proposed method enhances the module mapping process by ensuring that application modules are assigned to the most suitable fog devices. This mapping is achieved through the integration of AHP and TOPSIS algorithms, where AHP determines the relative weights of evaluation criteria, ensuring a systematic and transparent weighting process. TOPSIS then facilitates optimal placement by ranking the available fog nodes according to their closeness to the best solution.

This study presents a framework with a healthcare focus that comprises an initial classification step, in order to priorities critical healthcare applications according to emergency level. Additionally, this approach enhances cloud-fog resource optimization and improves the effectiveness of application placement.

To illustrate this process, Figure 4 provides a flowchart that outlines the key steps:

- Input of parameters: Initial data, such as application requirements and resource availability, are provided.
- AHP execution: Criteria weights are calculated using pairwise comparisons.
- TOPSIS execution: Fog nodes are ranked based on their performance across the weighted criteria.
- Check of resources: Depending on the rating and resource constraints, applications are either deployed on the cloud or the best fog node.

In the following section, we provide a step-by-step explanation of the AHP and TOPSIS algorithms, including their mathematical formulations and implementation details.

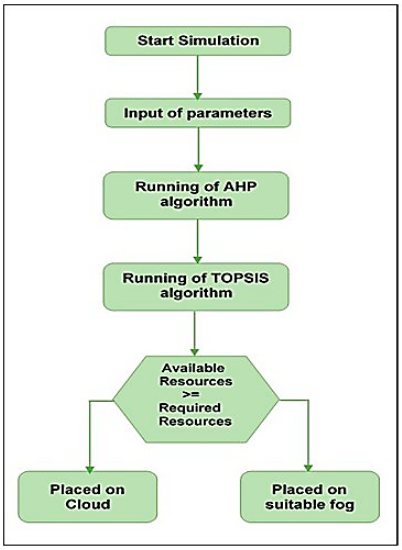


Figure 4. Flowchart of proposed algorithm based on AHP and TOPSIS.

3.5.1 Selection of Criteria

The selection of the evaluation criteria—CPU power, RAM, and network bandwidth—was based on their critical impact on determining the performance and suitability of fog nodes in cloud-fog environments, and Latency was selected based on its critical impact for real-time healthcare applications.

Each criterion directly impacts the quality of service (quality of service) and resource efficiency. Below, we provide a brief justification for their inclusion:

- CPU: This parameter indicates a fog node's computational capacity, playing a critical role in executing resource tasks and ensuring the efficient processing of applications.
- RAM: The memory capacity of a fog node determines its ability to support applications. Sufficient RAM availability is essential for maintaining system stability and responsiveness.
- Latency: This metric quantifies the delay in data transmission, which is particularly crucial for real-time and time-sensitive applications. Minimizing latency enhances system responsiveness and improves the overall user experience. One of the primary advantages of fog computing is its ability to reduce latency by processing data closer to the source.
- Network Bandwidth: Determines the data transfer capacity of a node, which is vital for supporting data-intensive applications and maintaining seamless communication in distributed systems.

3.5.2 Analytic Hierarchy Process (AHP)

The AHP is a structured method that uses mathematical principles to organize and analyze complex decisions (Majumder, 2015). AHP involves three key steps: hierarchy structuring, relative measurement (which includes pairwise comparisons between criteria and alternatives), and distributive synthesis (where priorities are normalized to sum to 1) (Salomon et al., 2016).

The AHP process consists of the following steps (Velmurugan et al., 2011):

- Determination of alternative and criteria (in this proposed work: Fog devices are regarded as alternatives, and diverse characteristics [CPU, RAM, UpBw, DownBw, and Latency] are viewed as criteria).
- Creating a pairwise comparison matrix (PCM) by determining the relative significance of performance criteria and assigning weightings accordingly using a 9-point semantic Saaty scale (Saaty, 2008) to determine the coefficients, as can be observed in Table 3. The proposed work generates the PCM as depicted in Table 4.
- Calculate the priority weight for each alternative for each criterion: Mathematically, the vector of priorities can be calculated as follows:

$$w_i = \frac{1}{n}\sum_{j=1}^{n}\frac{a_{ij}}{\sum_{i}^{n} a_{ij}} \quad i,j = 1,2,,,,n \tag{1}$$

$$\text{avec} \quad n : \text{nbr} \quad \text{of} \quad \text{criteria} \quad \text{and} \quad \sum_{i=1}^{n} \omega_i = 1$$

Table 3. The scale values used in a pairwise comparison matrix (Saaty, 2008).

Scale Value	Description
1	Equally Important
2	Weak or slight Importance
3	Moderately Important
4	Moderate plus
5	Strongly Important
6	Strong plus
7	Very Strongly Important
8	Very very strong
9	Extremely Important

Table 4. Pairwise comparison matrix of proposed work.

Criteria	CPU	RAM	UpBw	DownBw	Latency
CPU	1	3	6	5	6
RAM	0.33	1	6	5	6
UpBw	0.16	0.16	1	0.33	3
DownBw	0.2	0.2	3	1	3
Latency	0.16	0.16	0.33	0.33	1

- Calculate the Consistency Ratio (CR): To determine CR, it is essential to follow these three steps: 1- Begin by calculating the Eigen value (λ_{max}): Start by multiplying the right side of the matrix by the priority vector; Next, divide each element of the weighted sum matrix or the new vector by its respective priority vector element; Finally, calculate the average of these values to obtain λ_{max}.. 2- Next, calculate the consistency index (CI). 3- Lastly, determine the consistency ratio (CR) using the formula provided:

$$\mathrm{CI} = \frac{(\lambda_{max} - \mathrm{n})}{(\mathrm{n}-1)} \tag{2}$$

$$\mathrm{CR} = \frac{\mathrm{CI}}{\mathrm{RI}} \tag{3}$$

Choose the suitable random index (RI) value for a matrix by referring to Table 5.

If CR<0.1, the calculated weights are considered acceptable; however, if CR>0.1, it indicates the pairwise comparison matrix is inconsistent. It is advisable to re-evaluate and refine the weightings to achieve a consistent matrix.

After the consistency calculation, the priority weights vector (ω_i) will be used in the TOPSIS method to assign the weights to the criteria in the decision-making process.

Table 5. Random index of analytic hierarchy process (Velmurugan et al., 2011)

Size of matrix (n)	Random Index (RI)
1	0
2	0
3	0.58
4	0.9
5	1.12
6	1.24
7	1.32
8	1.41
9	1.45
10	1.49
11	1.51
12	1.58

3.5.3 Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) Algorithm

TOPSIS is a multicriteria decision analysis method that is Relies on the principle that the chosen solution should have the minimum distance from the ideal (positive ideal) solution and the maximum distance from the anti-ideal (negative ideal) solution. The steps involved in TOPSIS can be outlined as follows (Moghassem, 2010):

- Calculate the normalized decision matrix:

$$\boldsymbol{n_{ij}} = \frac{\boldsymbol{x_{ij}}}{\sqrt{\sum_{j=1}^{m} \boldsymbol{x_{ij}^2}}} \quad \mathrm{j} = 1 \ldots . \mathrm{m}, \mathrm{i} = 1 \ldots . . \mathrm{n} \tag{4}$$

- Calculate the weighted normalized decision matrix:

$$\mathbf{v_{ij}} = \mathbf{w_i} * \mathbf{n_{ij}} \quad \mathrm{j} = 1 \ldots . \mathrm{m}, \mathrm{i} = 1 \ldots . . \mathrm{n} \tag{5}$$

- Determine the positive ideal and negative ideal solution:

$$\begin{aligned} \mathrm{A}^{+} &= \left\{\mathrm{v}_1^{+} \ldots . . \mathrm{v}_\mathrm{n}^{+}\right\} = \begin{Bmatrix} (\max_\mathrm{j} \mathrm{v_{ij}} | \mathrm{i} \in \mathrm{I}), \\ (\min_\mathrm{j} \mathrm{v_{ij}} | \mathrm{i} \epsilon \mathrm{J}) \end{Bmatrix} \\ \mathrm{A}^{-} &= \left\{\mathrm{v}_1^{-} \ldots . . \mathrm{v}_\mathrm{n}^{-}\right\} = \begin{Bmatrix} (\min_\mathrm{j} \mathrm{v_{ij}} | \mathrm{i} \in \mathrm{I}), \\ (\max_\mathrm{j} \mathrm{v_{ij}} | \mathrm{i} \epsilon \mathrm{J}) \end{Bmatrix} \end{aligned} \tag{6}$$

Where "I" corresponds to benefit criteria and "J" corresponds to cost criteria.

- Calculate the separation measure using the n-dimensional Euclidean distance:

$$\begin{aligned} \mathrm{d}_\mathrm{j}^{+} &= \sqrt{\left\{\sum_{\mathrm{i}=1}^{\mathrm{n}} \left(\mathrm{v_{ij}} - \mathrm{v}_\mathrm{i}^{+}\right)^2\right\}} \\ \mathrm{d}_\mathrm{j}^{-} &= \sqrt{\left\{\sum_{\mathrm{i}=1}^{\mathrm{n}} \left(\mathrm{v_{ij}} - \mathrm{v}_\mathrm{i}^{-}\right)^2\right\}} \\ &\text{with} \quad \mathrm{j} = 1 \ldots . \mathrm{m} \end{aligned} \tag{7}$$

- Calculate the relative closeness to the ideal solution.

$$R_i = \frac{d_j^-}{\left(d_j^+ + d_j^-\right)} \quad \text{with} \quad j = 1 \ldots . m \tag{8}$$

Since $d_j^+ \geq 0$ and $d_j^- \geq 0$, then clearly $R_i \epsilon [0, 1]$

In the end, the values of R_i for the alternatives are arranged in descending order, and the alternative with the highest R_i is assigned to the rank '1'. In contrast, the remaining alternatives are ranked accordingly.

3.5.4 Determining the level of emergency for a patient

The determination of a patient's emergency level relies on data from health monitoring sensors; in our case, these measure arterial blood pressure. This impacts the Gateways (patients') classification mechanism that prioritizes task execution.

Three categories are identified in this study: high (hypertension), at-risk (pre-hypertension), and normal, denoted by the integers (3, 2, and 1), respectively, as depicted in Table 6. We include this level to underscore the critical need for rapid intervention and appropriate action in emergencies.

Table 6. Blood Pressure Categories (Harvard Medical School, 2025).

Level	Systolic mm/Hg
Normal	<120
at-risk	120> and<139
high	>140

The dataset, sourced from (kaggle), pertains to data concerning blood pressure and its association with heart disease. The method relies on systolic blood pressure values at rest obtained from a heart disease dataset.

3.5.5 Pseudo-code of the Proposed Work

The pseudo code for the suggested application placement technique (Algorithm 1) is given in the following subsection. The system prioritizes crucial applications using an emergency-level factor, TOPSIS for fog node ranking (Algorithm 3), and AHP for criterion weighting (Algorithm 2). To ensure a seamless, efficient decision-making process in the fog computing environment, the pseudo-code outlines the logic for determining the optimal placement.

3.6 Security and Privacy Considerations

In this study, we assume the architecture is secure and that the necessary security measures are configured to address privacy and security issues when managing sensitive medical data across cloud and fog environments. Techniques such as encryption protocols (including AES-256 (Advanced Encryption Standard with a 256-bit key) for stored data and TLS/SSL (Transport Layer Security / Secure Sockets Layer) for data in transit), secure data handling procedures (e.g. as cryptographic hash functions to verify data integrity), and access control mechanisms (e.g., RBAC (role-based access control) to limit unauthorized access), and the integration of Physical Unclonable Function (PUF) technology to enhance the security of IoT devices could be employed to ensure data confidentiality, integrity, and availability. Although the main focus of this work is on application placement and decision-making, we recognize the critical importance of security in healthcare apps.

Algorithm 1 AHP-TOPSIS Application Placement Policy

Require: Set of applications (Processing modules) A, set of Fog Nodes FNs, evaluation criteria W (CPU, RAM, Network Bandwidth, Latency)
Ensure: Optimal placement of applications on Fog Nodes
Step 1: Determine AHP Weights:
Derive the weight vector W using the AHP method.
Step 2: Apply TOPSIS for Node Selection:
Identify the best node for placement using the TOPSIS method.
Step 3: Determine Emergency Level and Classify Application:
For each application, compute the emergency level and classify them, prioritizing the Critical apps .
Step 4: Place Application Modules
for each module $m \in A$ **do**
    Identify the best Fog Node fn^* using TOPSIS.
    if resources in fn^* are sufficient **then**
        Place module m on fn^*.
        Update the resource availability of fn^*.
    else
        Place module m on the cloud.
        Update the cloud resource availability.

Algorithm 1.

4. Simulation Result and Discussion

To assess the proposed method, iFogSim is chosen. iFogSim is a tool for simulating IoT and Fog environments, enabling the measurement of the impact of resource management on metrics such as latency, energy consumption, network congestion, and cost. IFogSim is an extension of CloudSim and therefore inherits a variety of

features from it (Gupta et al., 2017). It is coded in Java and necessitates the Java Development Kit (JDK).

Algorithm 2 AHP Weight Calculation with Consistency Check

Require: Pairwise comparison matrix P of size $n \times n$
Ensure: Weight vector W of size n

Step 1: Calculate Column Sums of P
for each column j in P **do**
  $S[j] \leftarrow \sum_{i=1}^{n} P[i][j]$
Step 2: Normalize the Matrix
for each element $P[i][j]$ in P **do**
  $N[i][j] \leftarrow P[i][j]/S[j]$
Step 3: Calculate the Criteria Weights
for each row i in N **do**
  $W[i] \leftarrow \frac{\sum_{j=1}^{n} N[i][j]}{n}$ ▷ Criteria Weights
Step 4: Calculate Consistency Matrix
for each element $C[i][j]$ in P **do**
  $C[i][j] \leftarrow P[i][j] \times W[j]$ ▷ Consistency Matrix
Step 5: Calculate Weighted Sums
for each row i **do**
  $WS[i] \leftarrow \sum_{j=1}^{n} C[i][j]$ ▷ Weighted Sum
Step 6: Calculate Consistency Ratios
for each element i **do**
  $RS[i] \leftarrow WS[i]/W[i]$ ▷ Ratio Sum
$\lambda_{max} \leftarrow \frac{\sum_{i=1}^{n} RS[i]}{n}$
$CI \leftarrow (\lambda_{max} - n)/(n-1)$
$CR \leftarrow CI/RI[n]$ ▷ Predefined random index
Step 7: Check Consistency
if $CR < 0.1$ **then**
  Return W ▷ Consistent Weights
else
  Error: Adjust Pairwise Comparisons

Algorithm 2.

Algorithm 3 TOPSIS Method for Node Selection

Require: Decision matrix D of size $n \times m$, Weight vector W of size m
Ensure: Ranked list of alternatives

Step 1: Calculate Normalized Decision Matrix
for each criterion $j = 1$ to m **do**
  $S[j] \leftarrow \sum_{i=1}^{n} D[i][j]^2$
for each element $D[i][j]$ **do**
  $N[i][j] \leftarrow \frac{D[i][j]}{\sqrt{S[j]}}$
Step 2: Calculate Weighted Normalized Matrix
for each element $N[i][j]$ **do**
  $V[i][j] \leftarrow N[i][j] \times W[j]$
Step 3: Determine Ideal and Negative Ideal Solutions
for each criterion j **do**
  $A^{+}[j] \leftarrow \max V[:, j]$ ▷ Ideal Best
  $A^{-}[j] \leftarrow \min V[:, j]$ ▷ Ideal Worst
Step 4: Calculate Euclidean Distances
for each alternative i **do**
  $S^{+}[i] \leftarrow \sqrt{\sum_{j=1}^{m} (V[i][j] - A^{+}[j])^2}$
  $S^{-}[i] \leftarrow \sqrt{\sum_{j=1}^{m} (V[i][j] - A^{-}[j])^2}$
Step 5: Calculate Performance Scores
for each alternative i **do**
  $C[i] \leftarrow \frac{S^{-}[i]}{S^{+}[i] + S^{-}[i]}$
Step 6: Rank Alternatives in Descending Order of C
Return Sorted list of alternatives by C

Algorithm 3.

The simulation setup was designed to reflect realistic healthcare scenarios, with the following configurations:

- Fog Node Specifications: Each fog node was configured with varying levels of CPU power and RAM, as detailed in Table 2, to represent the heterogeneity of fog nodes in real-world deployments.
- Network Conditions: The network was modeled with latency and bandwidth values, also specified in Table 2, to simulate both high-performance and resource-constrained environments.
- Patient Data Parameters: We used a blood pressure dataset to simulate patient monitoring in a healthcare context. The dataset consists of blood pressure readings, which allowed us to evaluate the framework's ability to handle healthcare applications effectively under controlled conditions.

This section shows the performance of the suggested work compared to the proposed work in (Baranwal et al., 2020) and (Mahmud et al., 2019).

4.1 Performance metrics

Specific performance metrics are considered to ensure equal comparisons.

4.1.1 Network Relaxation Ratio (NRR)

It is a metric used by the fog computing environment to monitor network congestion. The NRR is determined by considering the expected access rate (Ar) of an application and the round-trip time (Rtt) of the fog node where the application is situated.

$$(\mathrm{NRR})_{\mathrm{avg}} = \frac{1}{\mathrm{NAs}} \Sigma \frac{2}{\mathrm{Ar.Rtt}} \tag{8}$$

Where NAs is the Number of Applications Successfully placed.

In this, an elevated NRR value indicates a lower possibility of network congestion, while a lower NRR value indicates a higher possibility of network congestion. Therefore, in evaluating network congestion in fog computing environments, an elevated NRR value is considered more favorable, as it indicates a more effective reduction in congestion and potentially improves performance for IoT applications.

4.1.2 Resource Gain (RG)

It is a performance metric that measures a user's resource consumption in a fog computing environment. The RG is calculated based on the fog node's resource availability (Ra) and the application's expected resource requirements (Rr).

$$(\mathrm{RG})_{\mathrm{avg}} = \frac{1}{\mathrm{NAs}} \Sigma \frac{\mathrm{Ra}}{\mathrm{Rr}} \tag{9}$$

Where NAs is the Number of Applications Successfully placed.

In this, a higher RG value indicates that the fog nodes have greater resource availability relative to the IoT application's resource requirements. Therefore, in evaluating fog nodes for hosting IoT applications, a higher RG value is considered more desirable, as it indicates a better match between the fog nodes' resource availability and the IoT application's resource requirements.

4.1.3 Application Placement Time (APT)

It is the duration of time required to decide which fog nodes to place applications on. This measure is essential for evaluating how effectively the fog computing environment's application placement process is working.

A shorter APT is desirable as it indicates that the placement process is efficient and applications can be processed and executed more quickly.

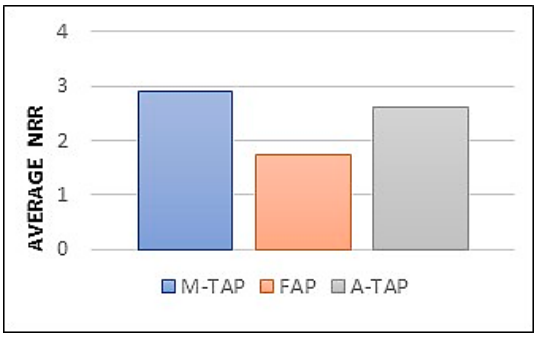
4.2 Discussion on results

In this case study, to provide a more precise representation of the proposed policy, a mediator within the FGL receives requests from 5 applications, and the FCL consists of 7 nodes, assuming that all fog nodes meet the minimum application requirements.

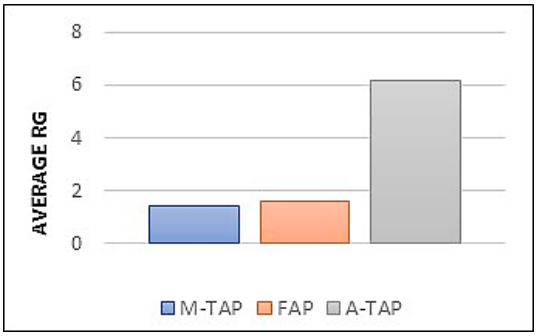
In the following discussion, we will explore our suggested application placement policy called A-TAP (AHP-TOPSIS Application Placement Policy). Furthermore, we will compare it with FAP, the Fuzzy Application placement policy outlined in (Mahmud et al., 2019), and M-TAP, the Modified TOPSIS Application placement policy suggested in (Baranwal et al., 2020).

Based on the performance metrics discussed, the values obtained are presented in Figure 5 below.

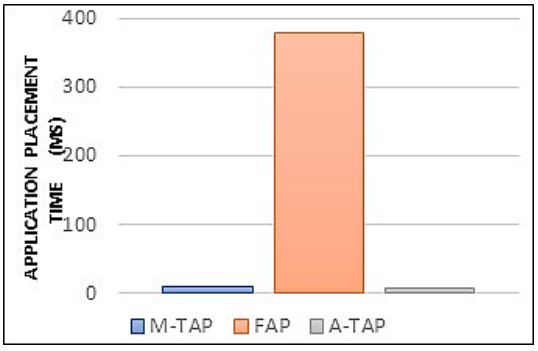
From Figure 5, first, regarding NRR, our suggested policy is better than FAP and slightly less important than M-TAP due to the use of a mediator. Moving on to RG, it has been enhanced compared to M-TAP and FAP due to task classification (emergency level) and resource reuse after processing completion. Lastly, regarding the time of placement, it is close to M-TAP but significantly lower than FAP.



a. NRR



b. RG



c. Application placement time.

Figure 5. Assessing the performance of the metrics.

In this test, we conducted 25 simulations and depicted the average outcomes in Figure 6. We varied the number of applications from 5 to 25 while keeping the number of fog nodes fixed at 7. This allowed us to evaluate the impact of the number of applications. As for the simulation parameters, we utilized random functions sourced from Table 7 provided below.

Table 7. Simulation parameters.

Parameter	Value
Expectation Metrics:	
Access rate	2-10 per sec
Resource requirement	1-8 CPU cores
Processing time	30-120 ms
Status Metrics:	
Round-trip time	100-600 ms
Resource availability	1-10 CPU cores
Processing speed	10-70 TIPS
Data size	1000-2000 instructions

From Figure 6, it is apparent that both average NRR and RG decrease as the number of applications increases. This is a logical outcome given the fixed resources and the increasing number of applications; significantly, the suggested method consistently outperforms both M-TAP and FAP.

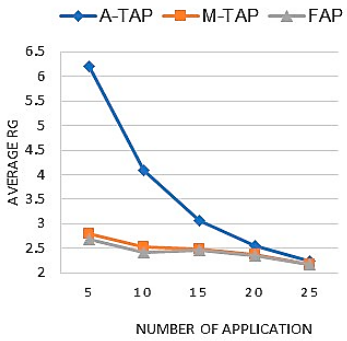
An experiment that focuses on an important parameter — application placement time— is shown in Figure 7. The observations are conducted through two tests. In the first test, the number of applications is varied from 5 to 50; in the second test, both the number of fog nodes (10 to 100) and the number of applications (5 to 50) are varied.

The results, as shown in Figure 7a, indicate that the placement time gradually increases with the number of applications. Referring to Figure 7 b, the results show that there is a corresponding increase in time with greater input capacity (number of applications, number of fog nodes). This increase is interesting since it is gradual, shows similarities in values between A-TAP and M-TAP, and is much less than FAP.

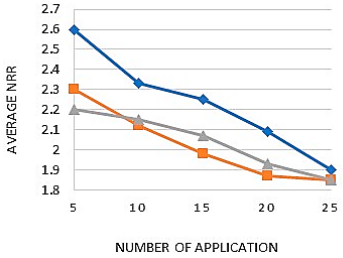
4.3 Latency and Energy consumption

In the IoT fog environment we focus on two key factors: latency and energy consumption are the most important metrics for evaluating a system's responsiveness.

Latency is the delay between the initiation of a request and the receipt of the desired response. Energy consumption refers to the amount of energy utilized by a system or device. This is especially important in the context of fog computing due to its direct impact on system efficiency and sustainability. In this study, energy consumption and latency were analyzed using the predefined functions provided by the iFogSim simulator. In such systems, minimizing both latency and energy consumption is critical, as they directly impact on the efficiency and responsiveness of healthcare applications.

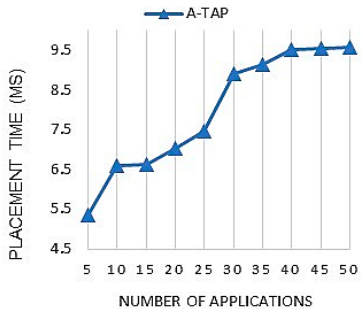


a. RG per Number of application.

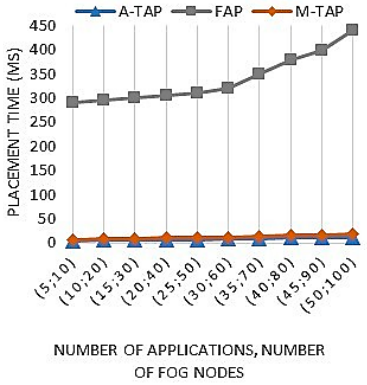


b. NRR per Number of application.

Figure 6. The performance of NRR and RG per application.



a. Varying the number of applications.



b. Varying the number of applications and Fog nodes.

Figure 7. Application placement time.

In our simulation, we increased the number of applications from 5 to 25 while maintaining a constant number of fog nodes at 7. This allowed us to obtain these latency and Energy consumption values. As seen in Figures 8 and 9, we recorded the values each time.

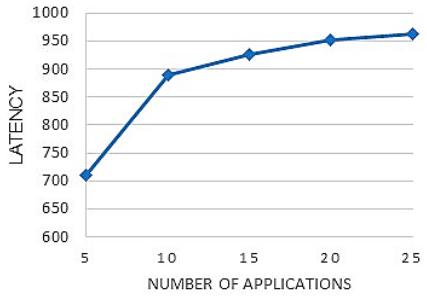


Figure 8. Latency per Number of applications.

It is evident from Figures 8 and 9 that these values are appropriate for this application. This data indicates that the application's responsiveness and efficiency are preserved, and its performance remains within reasonable limits. This shows that the tasks can be completed by the system efficiently and without any major issues. However, further work is required to optimize energy consumption in order to improve overall performance. To maintain the application's sustainability and efficiency, a thorough study of the patterns of current energy use will be conducted, and initiatives to lower power consumption will be put into practice.

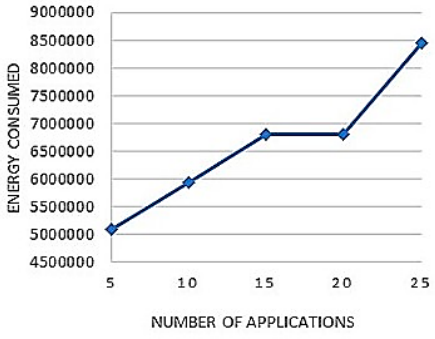


Figure 9. Energy Consumed per Number of Applications.

4.4 Limitations of the Proposed Framework

As the framework scales to support more fog nodes and applications, the increasing computational demands could slow down real-time decision-making, particularly in resource-limited environments. This becomes even more critical in real-time healthcare situations, where unexpected spikes in data —like during emergencies or outbreaks—might overload the system. To address these challenges, the framework could integrate real-time monitoring and adaptive resource allocation to scale resources in response to demand dynamically. Prioritizing critical applications and implementing fault-tolerant mechanisms would also improve reliability under unpredictable conditions. Tackling these issues is key to ensuring the framework works effectively in real-world, high-pressure healthcare scenarios.

5. Conclusions

In the fog computing environment, selecting an appropriate fog node is a significant challenge. In this study, our primary objective was to develop an application placement policy with a special focus on responding to patients' urgent needs. To determine which processing fog node would perform the best, we applied two multi-criteria decision-making techniques: AHP and TOPSIS. In addition, we considered the patients' emergency levels as a crucial reference point for categorizing fog gateways (patients). The experiment's outcomes indicate how well the suggested approach works and how well it fits the examined issue. Furthermore, by comparing it with the existing literature, the results of the experiment indicate that our approach not only performs well but also surpasses previous findings.

The proposed MCDM-based framework, while applied to patient monitoring in hospitals, can be extended to other healthcare domains such as telemedicine, ambulance management, and home-based patient monitoring. By integrating machine learning and predictive modeling, this approach could enhance real-time decision-making for resource management in remote healthcare settings. In telemedicine, it could optimize the placement of computational tasks to ensure low-latency video consultations and real-time diagnostics. In ambulance management, efficient application placement can enhance emergency response coordination by ensuring seamless data transmission among ambulances, hospitals, and dispatch centers. Additionally, in home-based patient monitoring, the framework could help balance computational load across fog nodes to provide continuous and reliable healthcare services. Beyond technical applications, this research also has implications for healthcare policy and resource allocation, particularly in resource-constrained settings. By optimizing resource utilization and reducing latency in critical applications, the proposed approach

can aid in decision-making for healthcare technology deployment.

In future research, several avenues exist to improve the framework. First, we can improve the system's ability to adapt in real time to changes in workload and network conditions. This would help it perform well even in unpredictable situations. Second, we can use machine learning to predict future resource needs and optimize where applications are placed. By analyzing historical data, the system could plan and allocate resources more efficiently. Another important step is to focus on energy efficiency and reliability, making sure the system uses resources wisely, especially in areas with limited resources. Finally, testing the framework in real-world healthcare settings would help us understand how well it works in practice and identify areas for improvement. These changes would make the framework more effective and ready for real-world use.

Conflict of interest

The authors have no conflict of interest to declare.

Funding

The authors received no specific funding for this work.

Acknowledgements

The authors wish to acknowledge the University of Abou Bekr Belkaid Tlemcen.

References

(s.d.). Consulté le 2025, sur kaggle: https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction

Al-Tarawneh, M. A. (2022). Bi-objective optimization of application placement in fog computing environments. *Journal of Ambient Intelligence and Humanized Computing*, *13*(1), 445-468. https://doi.org/10.1007/s12652-021-02910-w

Ahuja, S. P. & Deval, N. (2021). From Cloud Computing to Fog Computing: Platforms for the Internet of Things (IoT). In I. Management Association (Ed.), *Research Anthology on Architectures, Frameworks, and Integration Strategies for Distributed and Cloud Computing* (pp. 999-1010). IGI Global Scientific Publishing. http://dx.doi.org/10.4018/978-1-7998-5339-8.ch047

Baranwal, G., & Vidyarthi, D. P. (2021). FONS: a fog orchestrator node selection model to improve application placement in fog computing. *The Journal of Supercomputing*, *77*(9), 10562-10589. https://doi.org/10.1007/s11227-021-03702-x

Baranwal, G., Yadav, R., & Vidyarthi, D. P. (2020). QoE aware IoT application placement in fog computing using modified-topsis. *Mobile Networks and Applications*, *25*(5), 1816-1832. https://doi.org/10.1007/s11036-020-01563-x

Ghobaei-Arani, M., Souri, A., & Rahmanian, A. A. (2020). Resource management approaches in fog computing: a comprehensive review. *Journal of Grid Computing*, *18*(1), 1-42. https://doi.org/10.1007/s10723-019-09491-1

Gupta, H., Vahid Dastjerdi, A., Ghosh, S. K., & Buyya, R. (2017). iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments. *Software: Practice and Experience*, *47*(9), 1275-1296. https://doi.org/10.1002/spe.2509

Harvard medical school. (s.d.). Consulté le 03 08, 2025,sur https://www.health.harvard.edu/blog/new-high-blood-pressure-guidelines-2017111712756

Goudarzi, M., Wu, H., Palaniswami, M., & Buyya, R. (2020). An application placement technique for concurrent IoT applications in edge and fog computing environments. *IEEE Transactions on Mobile Computing*, *20*(4), 1298-1311. https://doi.org/10.1109/TMC.2020.2967041

Lera, I., Guerrero, C., & Juiz, C. (2019, June). Analyzing the applicability of a multi-criteria decision method in fog computing placement problem. In *2019 Fourth International Conference on Fog and Mobile Edge Computing (FMEC)* (pp. 13-20). IEEE. https://doi.org/10.1109/FMEC.2019.8795361

Mahmud, R., Ramamohanarao, K., & Buyya, R. (2018). Latency-aware application module management for fog computing environments. *ACM Transactions on Internet Technology (TOIT)*, *19*(1), 1-21. https://doi.org/10.1145/3186592

Mahmud, R., Srirama, S. N., Ramamohanarao, K., & Buyya, R. (2019). Quality of Experience (QoE)-aware placement of applications in Fog computing environments. *Journal of Parallel and Distributed Computing*, *132*, 190-203. https://doi.org/10.1016/j.jpdc.2018.03.004

Majumder, M. (2015). Multi criteria decision making. In *Impact of urbanization on water shortage in face of climatic aberrations* (pp. 35-47). Singapore: Springer Singapore. https://doi.org/10.1007/978-981-4560-73-3_2

Mishra, M. K., Ray, N. K., Swain, A. R., Mund, G. B., & Mishra, B. S. P. (2019). An adaptive model for resource selection and allocation in fog computing environment. *Computers & Electrical Engineering*, *77*, 217-229. https://doi.org/10.1016/j.compeleceng.2019.05.010

Moghassem, A. R. (2010). Application of TOPSIS approach on parameters selection problem for rotor spinning machine. *Fibers and Polymers*, *11*(4), 669-675. https://doi.org/10.1007/s12221-010-0669-7

Naha, R. K., Garg, S., & Chan, A. (2018). Fog computing architecture: Survey and challenges. *arXiv preprint arXiv:1811.09047*. https://doi.org/10.48550/arXiv.1811.09047

Redowan Mahmud, Satish Narayana Srirama, Kotagiri Ramamohanarao, Rajkumar Buyya,. (2019). Quality of Experience (QoE)-aware placement of applications in Fog computing environments,. *Journal of Parallel and Distributed Computing , 132*, 190-203. https://doi.org/10.1016/j.jpdc.2018.03.004

Natesha, B. V., & Guddeti, R. M. R. (2018, December). Heuristic-based IoT application modules placement in the fog-cloud computing environment. In *2018 IEEE/ACM international conference on utility and cloud computing companion (UCC Companion)* (pp. 24-25). IEEE. https://doi.org/10.1109/UCC-Companion.2018.00027

Mishra, S., Sahoo, M. N., Bakshi, S., & Rodrigues, J. J. (2020). Dynamic resource allocation in fog-cloud hybrid systems using multicriteria AHP techniques. *IEEE Internet of Things Journal*, *7*(9), 8993-9000. https://doi.org/10.1109/JIOT.2020.3001603

Varshney, S., Sandhu, R., & Gupta, P. K. (2021, October). QoE-based resource management of applications in the fog computing environment using AHP technique. In *2021 6th international conference on signal processing, computing and control (ISPCC)* (pp. 669-673). IEEE. https://doi.org/10.1109/ISPCC53510.2021.9609479

Saaty, T. L. (2008). Decision making with the analytic hierarchy process. *International journal of services sciences*, *1*(1), 83-98. https://doi.org/10.1504/IJSSCI.2008.017590

Salomon, V. A. P., Tramarico, C. L., & Marins, F. A. S. (2016). Analytic hierarchy process applied to supply chain management. In *Applications and Theory of Analytic Hierarchy Process-Decision Making for Strategic Decisions*. IntechOpen. http://dx.doi.org/10.5772/64022

Velmurugan, R., Selvamuthukumar, S., & Manavalan, R. (2011). Multi criteria decision making to select the suitable method for the preparation of nanoparticles using an analytical hierarchy process. *Die Pharmazie-An International Journal of Pharmaceutical Sciences*, *66*(11), 836-842. https://doi.org/10.1691/ph.2011.1034

Yousefpour, A., Fung, C., Nguyen, T., Kadiyala, K., Jalali, F., Niakanlahiji, A., ... & Jue, J. P. (2019). All one needs to know about fog computing and related edge computing paradigms: A complete survey. *Journal of systems architecture*, *98*, 289-330. https://doi.org/10.1016/j.sysarc.2019.02.009