A Dynamic Adaptive Bio-Inspired Multi-Agent System for Healthcare Task Deployment

Hamza Reffad
LRSD, Faculty of Technology, Technology Department, University Ferhat Abbas Sétif 1, Algeria
hamza.reffad@univ-setif.dz (corresponding author)

Adel Alti
LRSD, Faculty of Sciences, Computer Science Department, University Ferhat Abbas Sétif 1, Algeria
alti.adel@univ-setif.dz

Ahmed Almuhirat
Department of Management Information Systems, College of Business & Economics, Qassim University, Saudi Arabia
a.almuhirat@qu.edu.sa

Received: 16 December 2022 | Revised: 30 December 2022 | Accepted: 3 January 2023

ABSTRACT

The use of the Internet of Things (IoT) in healthcare is increasing significantly, bringing high-quality health services, but it still generates massive data with massive energy consumption. Due to the limited resources of fog servers and their impact on limiting the time needed for health data analysis tasks, the need to handle this problem in a fast way has become a necessity. To address this issue, many optimization and IoT-based approaches have been proposed. In this paper, a dynamic and adaptive healthcare service deployment controller using hybrid bio-inspired multi-agents is proposed. This method offers optimal energy costs and maintains the highest possible performance for fog cloud computing. At first, IGWO (Improved Grey Wolf Optimization) is used to initialize the deployment process using the nearest available fog servers. Then, an efficient energy-saving task deployment was achieved through Particle Swarm Optimization (PSO) to reduce energy consumption, increase rewards across multiple fog servers, and improve task deployment. Finally, to ensure continuous control of underloaded and overloaded servers, the neighborhood multi-agent coordination model is developed to manage healthcare services between the fog servers. The developed approach is implemented in the iFogSim simulator and various evaluation metrics are used to evaluate the effectiveness of the suggested approach. The simulation outcome proved that the suggested technique provides has better performance than other existing approaches.

Keywords-IoT; multi-agent; energy consumption; PSO; grey wolf optimization; fog-cloud

I. INTRODUCTION

We live in the era of the digital healthcare revolution. Every day, healthcare organizations generate billions of gigabytes of data from various sources of information such as medical sensors (IoT), social networks, telemedicine, etc. For this reason, data production and energy consumption are growing dramatically [1]. For example, a smart hospital consists of different actors (i.e. physicians, patients, and nurses) with various smart devices (i.e. smartphones, tablets), exchanging different health-related data (temperature and humidity levels, user glucose, user movement state, etc.), analysis results, and technical reports throughout various health services. Due to the explosion of services and massive energy consumption, the need to handle a large number of healthcare services and minimize their energy consumption increases sharply. Nowadays, most developers focus on improving healthcare task scheduling with optimum cost and reduced execution time. Interesting service configurations have been produced, satisfying the user requirements, but without considering the change of their needs and the incoming loads of applications on the limited resources of devices.

On the other hand, a great deal of data is collected through many different devices (wearables, medical/vital monitors, smartphones, etc.) and their heterogeneity is related to the type of data (text, video, image, sound) or the type of service such as healthcare services (glucose level, heart beating, etc.), spatiotemporal services (location, time, etc.), and social services (Facebook, Twitter, etc.). All these factors make service management more complex, and their placement is certainly very difficult [2]. Thus, many researchers work on service placement through a deployment component called service controller. They aim to deploy service on external data centers, build deployment scripts, and dynamically deploy it to the desired data centers [3].

Cloud computing is an interesting solution to efficiently deploy large candidate services. It allows interconnecting and integrating IoT devices and fogs according to the current services and users' needs. Besides, it is essential to know that the deployment of service is not a standardized solution. It can vary according to specific needs [4]. For this reason, service controllers must use advanced and intelligent techniques to deploy large candidates of services. In recent years, several approaches for efficient service deployment have been proposed and are generally classified into 3 main categories [5-19]: (1) green IoT approaches [5-9], (2) optimization approaches [11-14], and (3) hybrid approaches combining mobile devices, fog, and cloud optimization techniques [15-19]. Authors in [8] presented a new model called Health-Fog based on deep learning for saving energy on the fog and the cloud while the data are processed and transmitted from the fog to the cloud. Authors in [15] introduced a microservice-based system for Industrial Internet of Things (IIoT) in distributed fog cloud environments. Authors in [19] introduced a new priority traffic-conscious task placement algorithm that aimed to reduce the energy use of incoming IoT traffic servers. These techniques do not react to frequent moving scenarios of the users and none of them could get an optimum consensus between energy saving, processing load, computational cost, and fast waiting time. The most striking case is the health mobile applications where the mobility causes them to lose valuable time in providing accurate diagnosis results to their patients. There is a high computing required to select the right server to deploy services dynamically.

In the current work, a novel bio-inspired adaptive and dynamic multi-agent system for providing optimal deployment of healthcare services and applications under context changes (e.g. user's location, workload, bandwidth, reduced total time of execution) is presented. It enables much less deployment cost of health services in an optimal and context-aware fog cloud design and ensures effective solutions to deploy complex services by subdividing them into smaller ones. The main contributions of the current paper are:

- Providing an optimal deployment map for incoming loads of healthcare applications using a hybrid bio-inspired system by combining the Improved Grey Wolf Optimization (IGWO) and Particle Swarm Optimization (PSO) in IoT fog-cloud systems.
- Developing an adaptive and dynamic multi-agent-based controller that offers redeployment map for the ongoing needs of various users to the corresponding neighboring fog servers while minimizing energy consumption and waiting time, maximizing the gain of service.
- Evaluation of the proposed approach on the real healthcare data set of Qassim hospital in terms of energy consumption and execution time using the iFogSim simulator.

II. THE CASE STUDY: QASSIM SMART HOSPITAL

Qassim Hospital consists of many floors. Each floor consists of many patient rooms and a fog server. Each room is equipped with sensors (e.g. IP cameras). Any patient can be equipped with biosensors (glucose meter, heart tracking, smart bracelet, and GPS) that monitor medical data. The monitored information covers patient identifier, glucose level, temperature, patient location, etc. The monitored information is collected in a gateway. The aggregated context health data of the patient in a given room are transmitted to the fog server and the cloud controller for data analysis and storage. The local fog server triggers an immediate response to the device that gives an order to release some quantity of the drug in the body of a patient. The IoT healthcare application consists of connected IoT and various health tasks. Each task is defined by arrival time, priority, deadline, service time, and requested resources. Connected things are constantly consuming energy and sending huge data flows. This leads to low latency and consumer pressure on the amount of data flow on the fog and the Internet. Whether for patient healthcare, examinations or treatments, or hospitalization, an optimized task deployment approach of data collected by increasingly smart sensors can shed new light, improve flows, and reduce energy consumption. This case study raises several needs: (1) ensuring an optimal deployment map for incoming loads of healthcare applications and (2) determining their (re-) deployment into fog-cloud systems while reducing energy consumption and minimizing waiting time.

III. THE PROPOSED APPROACH

To develop an efficient and effective task deployment approach that can provide minimum energy costs and maximum benefits for users (i.e. patients, medical personnel, and administrators), we propose a new adaptive and dynamic system based on bio-inspired and cooperative agents with capacities like mobility, scalability, and efficiency. Figure 1 shows an overview of the proposed approach. Our goal is to ensure the coordination between the different agents of the healthcare system to determine the nearest best fog server between all feasible servers and select the server with the lowest processing load, waiting tasks, and energy consumption.

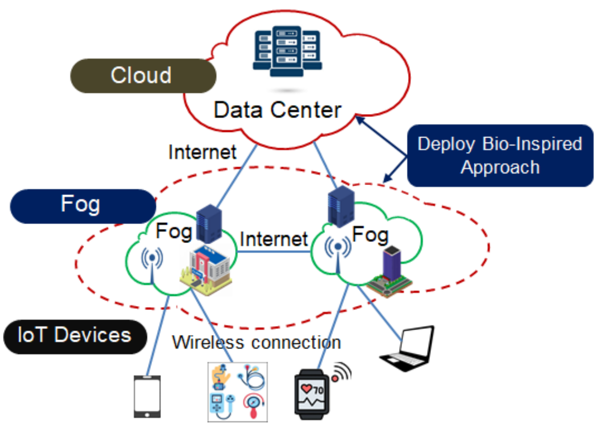

Fig. 1. An overview of the proposed approach.

The proposed approach consists of 3 static agents (Interact Agent, Deployer Agent, and Execution Agent) and two mobile agents (Monitor Agent and Bio-Research Agent) that interact and cooperate for realizing the deployment process. When a user sends a request, it consists of an application deployment to Interact Agent who plays the role of a mediator. Bio-Research Agent is a mobile bio-inspired agent that is able to migrate to fog servers to look for the requested deployment application. The Bio-Research Agent searches for local fog servers able to make partial or full deployment of the requested application. The Bio-Research Agent gives all the possible solutions and sends its result to the Deployer Agent. The Deployer Agent selects the best solution (a single fog server or a set of fog servers). Various politics based on weights (user's location, CPU load, energy-saving, high bandwidth) are used. Once the deployment map has been generated and the application has been deployed, the deployment results are sent back to the user. If the Bio-Research Agent does not find any solution on local fogs, it sends requests to the Deployer Agent to deploy tasks involved in the application on cloud servers. The Monitor Agent is responsible for handling the environment context during the execution of tasks such as changes of users' needs and incoming new urgent tasks. If it is about a single fog server, it is going to move and if it is about more than one fog servers, it clones himself, moves towards one fog server, and sends its clones to the other fog servers. The context execution parameters (CPU load, current bandwidth) will be transmitted to the Deployer Agent. The Deployer Agent selects the best fog server and generates the re-deployment map.

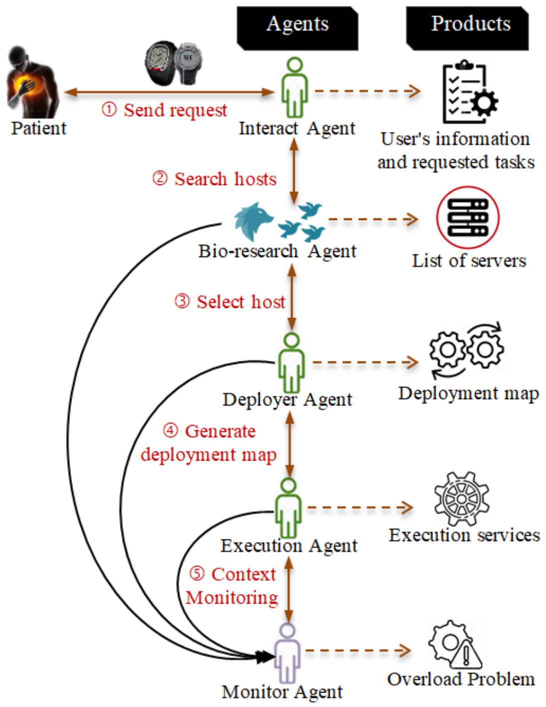

Fig. 2. The multi-agent functional model of AD-GWOPSO.

A. *Problem Formulation*

The problem consists of finding out an optimal deployment of IoT applications on IoT-Fog-Cloud infrastructure for minimizing energy consumption while maximizing the benefits of users (i.e. response time, reliability, and security). We consider three hierarchical levels of fog infrastructure $\mathcal{I}$ consisting of $\mathcal{N}$ physical nodes. The IoT layer (level 0) regroups a set of $\mathcal{T}$ smart devices, sensors, actuators, mobile devices, and edge devices. The fog computing layer (level 1) is a set of $\mathcal{F}$ fog servers and getaways. The cloud computing layer (level 2) consists of a set of $\mathcal{C}$ large-scale data centers.

The physical network is denoted as $\mathcal{G}_{\mathcal{I}} = (\mathcal{N}, \mathcal{E})$ which consists of a set of nodes (*i.e. servers*) $\mathcal{N}$ and a set of edges $\mathcal{E}$. Each edge $e_{i,j}$ represents a link between a server i and a server j. Each node s_i ($1 \le i \le \mathcal{N}$) is localized by a position (x_{s_i}, y_{s_i}). A node s_i has a queue q_i defined by the occupation rate ρ_i and maximum size max_q_i. Each node s_i has a current workload $W_{s_i}^{CPU}$. $E_{s_i}^{CPUa}$, denotes the power consumed by the server s_i in the active state and $E_{s_i}^{CPUidle}$ denotes the power consumed in the idle state. BW_{ij} represents the available bandwidth and TP_{ij} the data transferred (throughput) between servers i and j. The energy bandwidth consumed between servers i and j in their active state is E_{ij}^{BWa}, and the energy bandwidth consumed between i and j in the idle state is E_{ij}^{BWidle}. To solve the problem, we use a four step approach. At first, we sort all tasks by priority. A task is designed with a name and its priority value. Then, we initiate and sort all possible solutions (i.e. servers' nodes) by their weights based on the user's location and resource availability. Thirdly, we lunch a bio-inspired search algorithm that gives all possible server chains providing the deployment. The best solution is based on the weight (CPU charge, energy-saving, low bandwidth, and security) of each deployment. Finally, we ensure an automated task transfer between neighboring nodes for incoming loads of applications. The problem is formalized as follows:

- The set $U = \{u_1, u_2, \ldots, u_M\}$ of M users. Each user U_i requests the deployment of the application a_i and broadcasts his current location is (x_{u_i}, y_{u_i}).
- Each task t_i ($1 \le j \le$ M) can be deployed to a device S_k ($j = 1, 2, \ldots N$) by the projection $\mathcal{P} : T \; x \; U \rightarrow S$. This projection is called the deployment map.
- $TT_{s_i}^{t_j}$ is the time taken by a server s_i to perform a task t_j.

Equation (1) calculates the Euclidian distance separating the person's location from m servers of deployment map $\mathcal{P}$:

$$Dist(\mathcal{P}) = \sum_{i=1}^{m} \sqrt{\left(x_u - x_{s_i}\right)^2 + \left(y_u - y_{s_i}\right)^2} \qquad (1)$$

where x_u, y_u are the current location coordinates of the user u and x_{s_i},y_{s_i} are the coordinates of the fog server s_i. Fogs with the shortest distance closer to a person's current device tend to be selected as best fogs by minimizing $Dist(\mathcal{P})$. The best fog has the smallest distance from the person. Equation (2) calculates the total consumed computing energy E_{CPU} of the deployment map:

$$E_{CPU}(\mathcal{P}) = \sum_{s_i \in \chi} W_{s_i}^{CPU} \times E_{s_i}^{CPU_a} + \left(1 - W_{s_i}^{CPU}\right) \times E_{s_i}^{CPU_{idle}} \quad (2)$$

By reducing E_{CPU}, lower computing energy fogs tend to be chosen as the best deployment maps. Equation (3) calculates the total consumed network energy to transfer r tasks between fogs of deployment map $\mathcal{P}$:

$$E_{BW}(\mathcal{P}) = \sum_{(i,j)\in\mathcal{E}} TP_{ij} \times E_{ij}^{BW_a} + \left(BW_{ij} - TP_{ij}\right) \times E_{ij}^{BW_{idle}} \quad (3)$$

Equation (4) calculates the total network energy consumption, the total CPU power consumption applied to the fog servers for the task requests on deployment map:

$$f_{energy} = E_{CPU}(\mathcal{P}) + E_{BW}(\mathrm{P}) \quad (4)$$

Equation (5) is used to minimize the waiting time of the performed task.

$$f_{wt} = T_{wait} = \sum_{i=1}^{m} (\rho_i) \quad (5)$$

where ρ_i is the occupancy rate of fog i. Equation (6) is used to maximize the reward time of the servers:

$$f_{\text{gain}} = G_{\text{rewards}} = \sum_{i=1}^{m} \text{rewards} \times \mathrm{e}^{-\gamma \times TT_{s_i}^{t}} \quad (6)$$

where γ is the discount rate, ranging between 0 and 1.

Our goal is to establish an objective function (f) such that: (1) the total energy is minimized, (2) the global occupation rate of a person's tasks is minimized, and (3) the global gain of the fog server is maximized. From such a perspective, we define the objective function as follows:

$$f = \frac{w_{gain} \times f_{\text{gain}}}{w_{energy} \times f_{\text{energy}} + w_{time} \times f_{\text{wt}}} \quad (7)$$

where w_{energy} w_{time}, and w_{gain} are the weights associated with the objectives energy, time, and gain, respectively. All are normalized in [0, 1]. The higher the weight value, the more important this criterion is. The objective function is customizable according to the different customer needs.

B. The Dynamic Adaptative Service Placement Algorithm

- **Solutions coding:** Each solution is represented by an array of *n* fog servers.
- **Generate the initial population based on IGWO**: GWO [20] is the most well-known optimization approach, and is used to generate the initial population (i.e. list of fog servers). The wolf agent moves towards the nearest server and provides information on server CPU load, available storage, and bandwidth communication. The best list of servers is assigned a high number of particles according to its objective functions whereas a good server has a higher fitness value than a bad one. We build the vector $W = [W_1, , \ldots, W_n$, where $W_i = \frac{U_i^r}{\sum_{i=1}^{n} U_i^r}$ is used to allocate particles for each server and U_i^r is the utility of the given fog server i based on r attributes that describe the fog. The number of particles for all services is the total number of particles multiplied by their normalized weight to better valorize the suitable server and the fitness of each particle is evaluated according to (7) of their related energy, time, and gain functions.
- **Fitness evaluation:** From the initial particles' position that defines the service placement map, a particle changes iteratively from one map to another with respect to velocity. The velocity is defined between two maps: the local best map (P_{Best}) and the global best map (G_{Best}). Fog servers with the best score (lowest energy consumption, lowest occupation rate, and highest rewards by server) are assigned a high-priority value.
- **Optimal deployment map generation.** The system generates an optimal deployment map to those selected fog servers.
- **Copy deployment map and service execution.** The generated deployment map in the previous phase will be copied to fog servers and deploy services.

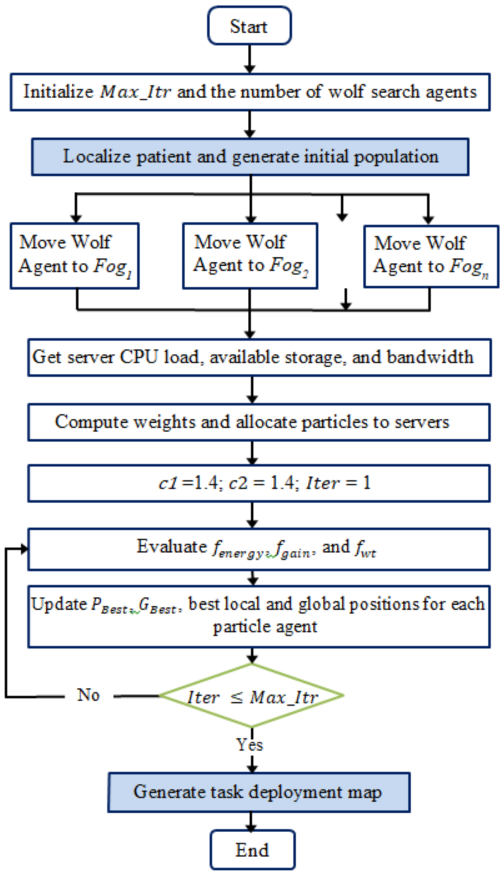

Fig. 3. Multi-agent functional model of the dynamic adaptive IGWO-PSO.

When unpredicted events occur, such as patient mobility or low latency on the fog server, the service controller will be switched to the adaptive mode, and the neighboring multi-agent technique is employed to tackle such events. Each service controller agent ag_S_i has a set of neighboring agents $ag\_S_i^j$. The agent ag_S_i of server S_i selects the best neighbors depending on occupancy rate, processing load, and transmission consumption energy. It checks the occupation rate ρ from a queue of the fog server S_i , and if it is greater than a threshold T_h then selects the tasks that are going to be transferred. The selection of a task is based on the type of task and the requested resources. A deployment request is created and sent to every neighbor. Everyone can accept or reject the request according to its occupation rate and processing load. The list of neighbors will be sorted by their queue size, available computing resources, and bandwidth. The controller transfers the requested tasks to the top ranked neighbors.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

The developed approach was implemented in Eclipse using iFogSim [21]. The number of rejected/waited tasks is used to assess the efficiency of the proposed approach. Various evaluation metrics were considered, including the number of rejected tasks, the number of waiting tasks, and the average makespan. The number of fog servers is 10 and the size of the waiting queue is 10. We simulated and compared different priority strategies: (1) Shortest Arrival Time (P-SAT) which selects the task with high priority and the shortest arrival time and (2) Least Requested Resources (P-LRR) that selects the task with high priority and the fewest requested resources.

A. Evaluation of Average Makespan

Figure 4 shows the average makespan time (ms) of the existing approaches (P-SPT, P-LRR), and the proposed approach with varying number of tasks. We can see that the proposed approach demonstrated optimum makespan compared to P-SPT and P-LPR due to the optimal deployment time using IGWO-PSO with a good energy saving while exploiting performance metrics.

B. Evaluating the Number of Rejected and Waited Tasks

Figure 5 shows the performance of the proposed approach with varying number of tasks from 87 to 2697. The number of rejected and waited tasks is computed for different numbers of tasks and different algorithms (P-SAT, P-LRR and proposed approach). The proposed approach obtained the optimum number of rejected tasks and the optimum number of waiting tasks for all the number of task values, due to the pre-selection of the best servers using IGWO and applying an adaptive task transfer to neighboring fog servers with a good server utility while exploiting performance metrics.

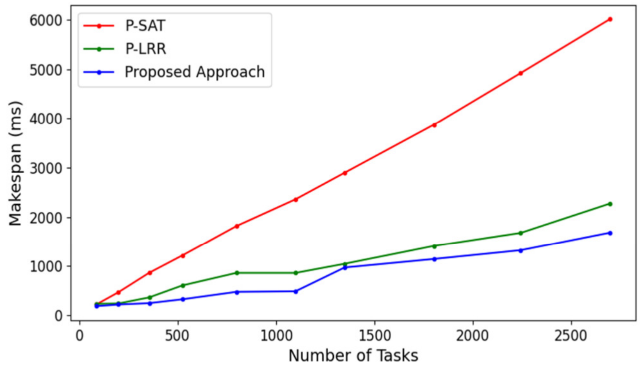

Fig. 4. The average makespan of the proposed and exisitng approaches.

C. Evaluation of Energy Consumption

To evaluate the energy consumption of our approach, we measure the average bandwidth/processing energy consumption of the proposed approach in cloud + fog and fog only. The proposed approach in fog environments is better in terms of bandwidth and processing energy consumption compared to the same approach in fog + cloud environments (Figure 6).

D. Discussion

In this paper, a dynamic and fast-to-deploy DS-IGWOPSO is proposed, one that ensures dynamic task deployment and applies a transfer to neighboring fog servers. Compared to other popular deployment algorithms (P-SPT and P-LRR), DS-IGWOPSO demanded less bandwidth and processing energy to deploy tasks. It also reduces the makespan average on varying number of tasks. To have IGWOPSO with its best performance, the neighbor selection should be based on the optimization of occupation rate and waiting time. However, it still depends on the number of requests to ensure optimal computational time.

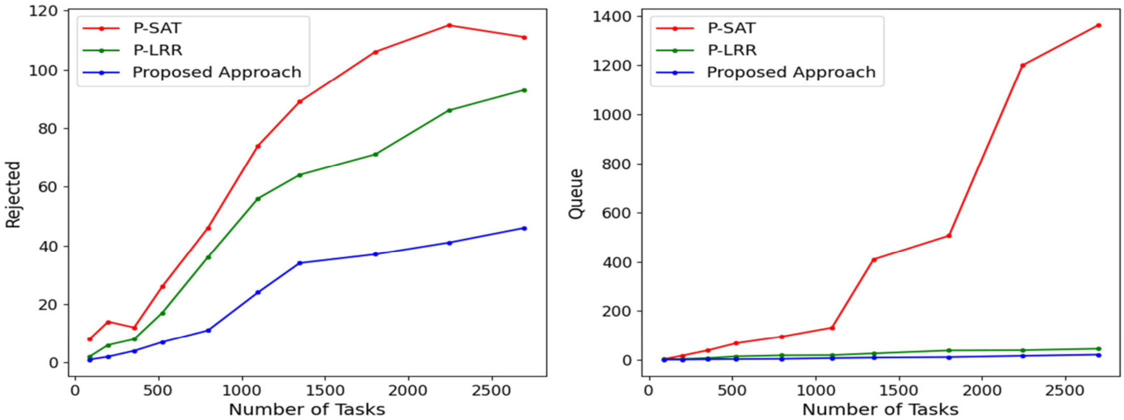

Fig. 5. Performance comparison.

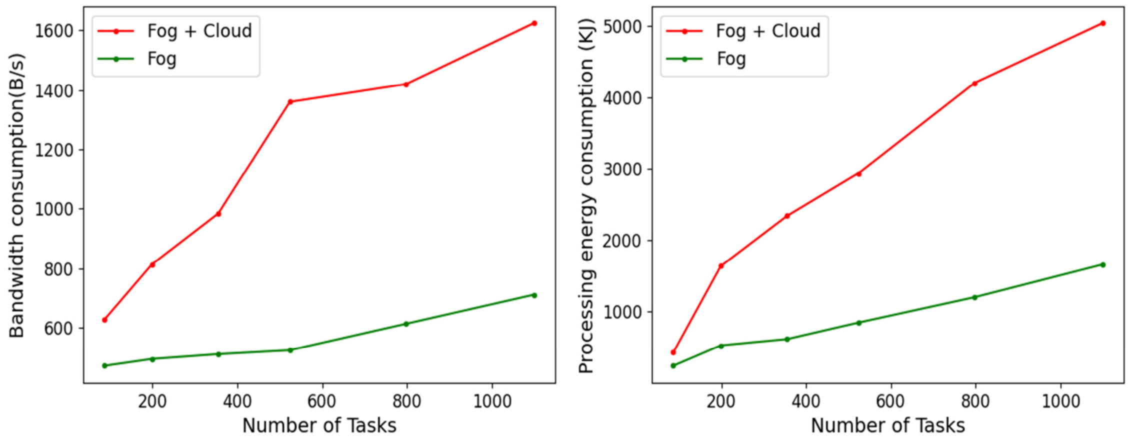

Fig. 6. Bandwith and processing energy consumption of the proposed approach in fog and fog+cloud.

V. CONCLUSION

An adaptive and dynamic bio-inspired multi-agent approach for deploying healthcare tasks in IoT-fog-cloud environments was developed and presented in this paper. Hybrid particle swarm and grey wolf optimizers were utilized for the determination of a set of optimal hosts when deploying health services to optimize deployment cost and improve response time at run-time. The approach is based on neighboring agents for ensuring the continued functioning of health services by using both service priority and high servers' resource of availability. The approach was evaluated by deploying different types of tasks. The evaluation of the obtained results was achieved through well-known metrics such as waiting tasks and average makespan time. The results show that the proposed approach is economic in terms of energy consumption. The obtained results are very encouraging and confirm the efficiency of the approach.

REFERENCES

[1] S. Kallam, R. Patan, T. V. Ramana, and A. H. Gandomi, "Linear Weighted Regression and Energy-Aware Greedy Scheduling for Heterogeneous Big Data," *Electronics*, vol. 10, no. 5, Jan. 2021, Art. no. 554, https://doi.org/10.3390/electronics10050554.

[2] H. H. A. Valera, M. Dalmau, P. Roose, J. Larracoechea, and C. Herzog, "DRACeo: A smart simulator to deploy energy saving methods in microservices based networks," in *2020 IEEE 29th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, Bayonne, France, Sep. 2020, pp. 94–99, https://doi.org/10.1109/WETICE49692.2020.00026.

[3] A. Adel, S. Laborie, and P. Roose, "Towards a Context-Aware Service and Quality Multimedia Adaptation for Healthcare Applications," in *International Conference on Digital Information Processing, E-Business and Cloud Computing*, 2013.

[4] S. Saxena and D. Saxena, "Green Cloud Computing Architecture with Efficient Resource Allocation System," *International Journal of Trend in Research and Development*, vol. 3, no. 6, pp. 248–251, 2016.

[5] H. H. lvarez-Valera, P. Roose, M. Dalmau, C. Herzog, and K. Respicio, "KaliGreen: A distributed Scheduler for Energy Saving," *Procedia Computer Science*, vol. 141, pp. 223–230, Jan. 2018, https://doi.org/10.1016/j.procs.2018.10.172.

[6] A. S. H. Abdul-Qawy, N. M. S. Almurisi, and S. Tadisetty, "Classification of Energy Saving Techniques for IoT-based Heterogeneous Wireless Nodes," *Procedia Computer Science*, vol. 171, pp. 2590–2599, Jan. 2020, https://doi.org/10.1016/j.procs.2020.04.281.

[7] S. Tuli *et al.*, "HealthFog: An ensemble deep learning based Smart Healthcare System for Automatic Diagnosis of Heart Diseases in integrated IoT and fog computing environments," *Future Generation Computer Systems*, vol. 104, pp. 187–200, Mar. 2020, https://doi.org/10.1016/j.future.2019.10.043.

[8] K. Haseeb, N. Islam, Y. Javed, and U. Tariq, "A Lightweight Secure and Energy-Efficient Fog-Based Routing Protocol for Constraint Sensors Network," *Energies*, vol. 14, no. 1, Jan. 2021, Art. no. 89, https://doi.org/10.3390/en14010089.

[9] A. A. Brincat, F. Pacifici, S. Martinaglia, and F. Mazzola, "The Internet of Things for Intelligent Transportation Systems in Real Smart Cities Scenarios," in *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, Limerick, Ireland, Apr. 2019, pp. 128–132, https://doi.org/10.1109/WF-IoT.2019.8767247.

[10] M. N. Hasan, R. N. Toma, A.-A. Nahid, M. M. M. Islam, and J.-M. Kim, "Electricity Theft Detection in Smart Grid Systems: A CNN-LSTM Based Approach," *Energies*, vol. 12, no. 17, Jan. 2019, Art. no. 3310, https://doi.org/10.3390/en12173310.

[11] N. Sasikaladevi and L. Arockiam, "Genetic Approach for Service Selection problem in Composite Web Service," *International Journal of Computer Applications*, vol. 44, no. 4, pp. 22–29, Apr. 2012, https://doi.org/10.5120/6252-8396.

[12] W. Song, W. Ma, and Y. Qiao, "Particle swarm optimization algorithm with environmental factors for clustering analysis," *Soft Computing*, vol. 21, no. 2, pp. 283–293, Jan. 2017, https://doi.org/10.1007/s00500-014-1458-7.

[13] F. Choukairy, "Optimization of energy consumption in a Cloud environment," Ph.D. dissertation, Laval University, Québec, QC, Canada, 2018.

[14] F. H. Khoso, A. Lakhan, A. A. Arain, M. A. Soomro, S. Z. Nizamani, and K. Kanwar, "A Microservice-Based System for Industrial Internet of Things in Fog-Cloud Assisted Network," *Engineering, Technology & Applied Science Research*, vol. 11, no. 2, pp. 7029–7032, Apr. 2021, https://doi.org/10.48084/etasr.4077.

[15] S. Omer, S. Azizi, M. Shojafar, and R. Tafazolli, "A priority, power and traffic-aware virtual machine placement of IoT applications in cloud data centers," *Journal of Systems Architecture*, vol. 115, May 2021, Art. no. 101996, https://doi.org/10.1016/j.sysarc.2021.101996.

[16] D. Mills, S. Sivarajah, T. L. Scholten, and R. Duncan, "Application-Motivated, Holistic Benchmarking of a Full Quantum Computing Stack," *Quantum*, vol. 5, Mar. 2021, Art. no. 415, https://doi.org/10.22331/q-2021-03-22-415.

[17] S. F. Issawi, A. A. Halees, and M. Radi, "An Efficient Adaptive Load Balancing Algorithm for Cloud Computing Under Bursty Workloads," *Engineering, Technology & Applied Science Research*, vol. 5, no. 3, pp. 795–800, Jun. 2015, https://doi.org/10.48084/etasr.554.

[18] M. E. Hassan and A. Yousif, "Cloud Job Scheduling with Ions Motion Optimization Algorithm," *Engineering, Technology & Applied Science*

Research, vol. 10, no. 2, pp. 5459–5465, Apr. 2020, https://doi.org/10.48084/etasr.3408.

[19] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, Mar. 2014, https://doi.org/10.1016/j.advengsoft.2013.12.007.

[20] H. Gupta, A. Vahid Dastjerdi, S. K. Ghosh, and R. Buyya, "iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments," *Software: Practice and Experience*, vol. 47, no. 9, pp. 1275–1296, 2017, https://doi.org/10.1002/spe.2509.