International Journal of System Design and Computing (IJSDC)
Volume 02, Issue 02, July - December (2024)

RESEARCH ARTICLE

DYNAMIC POWER ALLOCATION IN IOT-CLOUD ENVIRONMENT FOR HEALTHCARE APPLICATIONS

Kumarraja Andanapalli [1, *] and M. Suresh Kumar [2]

[1] Department of Electrical & Electronics Engineering, SRKR Engineering College, Bhimavaram, Chinamiram Rural, Andhra Pradesh 534204 India.

[2] Department of Computer Science and Engineering, Sri Venkateswara College of Engineering Sriperumbudur Chennai, 602117 India.

[*]*Corresponding e-mail: kumarraja@srkrec.ac.in*

Abstract – In the healthcare industry, the integration of the Internet of Things (IoT) and cloud computing (CC) enables access and sharing of worldwide health datasets. Thus, it solves complex problems like data security, privacy, and storage. However, the wide usage of cloud infrastructure increases traffic and reduces cloud performance. Hence, in this article, a novel hybrid Honey Pot-based Feed Feedback Neural System (HPbFFNS) framework was proposed to allocate resources optimally to medical applications (tasks). This framework incorporates the features of Honey pot optimization, and Feed Feedback Neural Network (FFNN). Initially, the health information of patients is collected using the IoT devices and forwarded into the gateway layer for further processing. The task scheduler in the gateway layer analyzes the resource availability, deadline, and priority of the incoming requests to reduce the response time, and waiting time. The honey pot fitness function in the resource allocator helps to allocate resources optimally to the tasks. Additionally, for verification purposes, the results are contrasted with those of current techniques. The experimental and comparative analysis confirms that the suggested model outperforms the traditional algorithms in terms of response time, energy consumption, and resource utilization.

Keywords *– Honey Pot Optimization, Feed Forward Neural System, Dynamic power allocation, Optimal Resource Allocation.*

1. INTRODUCTION

Cloud Computing (CC) is one of the developing techniques widely used in IoT-based applications because of its numerous advantages [1]. The basic principle behind CC is sharing of data/ information through the internet based on demand [2]. Cloud system contains hundreds of interconnected computers in a miscellaneous manner, where the data, files, and applications are accommodated [3]. CC incorporates parallel and distributed computing methods to provide sharing of resources like data/information, hardware, files, and software following the demand/ request on the cloud [4]. In the distributed system, CC offers a "Pay as you need" model [5]. Thus, the customer can access the computational software or platforms through the internet by paying the cost for the duration. Hence, the method eliminates the need to purchase of computational software or platforms for performing the task for the customers [6]. Moreover, the cloud system offers dynamic resource usage to the customers. In the cloud environment, virtual machines (VMs) are the processing units that share and compute the data/information dynamically as per the customer's demand/request [7]. Here, a huge number of VMs are interconnected and the resources are kept in a pre-emptive or non-pre-emptive manner resulting in the non-equal distribution of resources [8]. Thus, some VMs do not get the chance to share the resources. Moreover, when a task is assigned in the cloud, all VMs must implement the task faster to reduce computational time. In addition, all VMs must work in a parallel manner to minimize the time complexity [9]. This demands a proper scheduling algorithm to schedule the assigned task and complete the implementation within the available resources. When too many works are assigned to the VMs, they simultaneously work to complete the task [10].

While task assignation, the scheduler must check whether the VM is loaded or free. Moreover, it must assure that all the tasks are loaded to one VM leaving the other VMs extremely free or not utilized [11]. Thus, it is the responsibility of the scheduler to check whether the tasks are equally balanced in the cloud [12]. In recent times, one of the biggest challenges in CC is load balancing. To overcome this issue, an intelligent load-balancing mechanism is necessary for CC to safeguard the available resources [13]. Moreover, it improves the response time by implementing the assigned tasks faster [14]. Many researchers are conducted to develop an efficient load-balancing technique [15]. The major aim of the load-balancing strategy is to increase the response time of assigned tasks using the available resources [16]. The static load balancing method works properly when there are low fluctuations of load in VMs. Hence, it is not suitable for a cloud environment because the loads in the cloud vary unpredictably during the run time [17].

On the other hand, the dynamic load balancing strategy works properly at high fluctuating loads during the run time

[18]. Moreover, the high usage of the network and its resources demands an effective dynamic resource allocation (DRA) method [19]. Although various researchers were conducted to design an effective dynamic load balancing method, they face issues in time consumption, response time, and cost [20]. The existing techniques like deadline-based resource allocation algorithm [21], dynamic optimization mechanism [22], DRA approach based on particle swarm optimization (PSO) [23], etc., face challenges in cost and time complexity. Therefore, to overcome the existing techniques an optimized neural-based DRA system was designed in this article.

The key contribution of the presented work is defined below,

- Initially, the patient's health data is collected using the IoT devices in the IoT layer and forwarded into the IoT gateway.
- In the gateway layer, a hybrid HPbFFNS framework was designed to optimally allocate the resources in the IoT-cloud environment.
- The developed scheme estimates the deadline and priority of incoming requests to schedule the tasks optimally.
- The honey pot fitness function is integrated into the task allocator to determine the minimum resource availability in the cloud and to improve resource utilization.
- Finally, the performances of the presented model are evaluated and validated with existing techniques in terms of waiting time, response time, and energy consumption.

The presented article is sequenced as follows, the recent articles related to resource allocation (RA) are described in section 2, the problems of the existing techniques are illustrated in section 3, the proposed methodology is explained in section 4, the outcomes of the presented model are discussed in section 5, and the conclusion of the paper is mentioned in section 6.

2. RELATED WORKS

The following list includes a few recent studies on resource allocation.

In recent times, fog computing technology helps in supportive time-sensitive tenders related to the IoT. Conventionally, CC is widely used for processing IoT data. However, because of its high latency, it does not apply to time-sensitive applications. But RA in fog computing is a stimulating factor. Thus, Ranesh Kumar Naha et al. [21] presented a deadline-based RA algorithm to optimally allocate data to the users based on resource ranking. The developed scheme is employed in the CloudSim tool and the outcomes are determined. The performances of the developed scheme are validated in terms of processing time, delay in transmission, etc. However, the implementation cost is high in this method.

A fog computing network is one of the emerging techniques for the optimal sharing of information/ data in an IoT environment. However, allocating the resources optimally in the cloud environment is necessary to improve the response time. Thus, Zheng Chang et al. [22] presented a dynamic optimization mechanism for the CC network with multiple mobile devices to allocate resources optimally. This model utilizes hybrid radio and computational offloading method based on Lyapunov algorithm. Here, the main task is divided into several subtasks for reducing the complexity, and task load. Finally, the efficiency of the advanced model is authorized with experimental analysis.

The CC environment offers eliminates the need to purchase of computational software for performing specific tasks. But the high congestion that occurs in the network increases the latency, and cost. Hence, to overcome these issues D. Baburao et al. [23] developed a DRA approach based on PSO. This technique minimizes the task waiting time, latency, and power consumption, and increases the network's quality of service. Moreover, it provides better RA by eliminating the long-term inactive services from Radom Access Memory (RAM).

The incorporation of IoT technologies in industrial systems provides technical support. However, the heavy load congestion in the IoT-cloud environment reduces the response time and increases the waiting time. Therefore, Ying Chen et al. [24] suggested a load-balancing strategy based on the deep reinforcement technique. This developed model offers joint power control as well as DRA to mobile devices. Moreover, it minimizes the waiting time and latency in the network. Finally, the performances are estimated and evaluated by comparing them with traditional schemes.

Suchintan Mishra et al. [25] designed an effective RA system based on an analytic hierarchy process. The main aim of this method is to minimalize the latency acquired by each customer task. Generally, the cloud system is ineffective in analyzing latency-sensitive applications. This method employs a decision-making algorithm to provide load balancing in the cloud network dynamically. The experimental outcomes verify that the developed scheme outperforms the traditional methods. However, this RA system is not cost-efficient.

Nowadays, applications like business, mobile computing, and IT enterprises widely use CC platforms for storing and analyzing information. Hence, the resources such as CPU, input/output devices and memory can be used by the customers and charged as per the demand and usage. Therefore, J. Praveenchandar and A. Tamilarasi [26] presented an optimal power minimization method to enhance the effectiveness of RA. This utilizes a better task scheduling and prediction mechanism to offer optimal load balancing in the cloud network. This method overcomes the inability of the traditional scheme in offering optimal task scheduling and power consumption in CC. However, the latency parameter is not considered in this approach.

The cloud of things created by the incorporation of IoT and cloud environment increases the challenges in IoT and cloud areas. Hence, Seyedeh Maedeh Mirmohseni et al. [27] developed a load-balancing method based on the Markov model learning algorithm. In this model, the probability function is deployed to maximize network usage. The presented approach is simulated in the cloudsim environment and the results are evaluated.

However, the data loads are high in the cloud because of its wide usage. Traditional RA fails to improve response time and resource utilization. Hence, Hongbin Liang *et al* [28] developed an intelligent resource management strategy using the artificial intelligence mode. This model utilizes adaptive and intelligent schemes to offer dynamic resource scheduling in the cloud network. Moreover, it employs self-learning, and reinforcement learning to reduce the waiting time which occurs because of the large processing of data.

Amir Javadpour *et al* [29] developed a RA system for peer-to-peer networks and IoT. This model utilizes a deep neural framework for training the system to offer dynamic load balancing. The deep neural-based technique can balance huge loads at high fluctuations. Moreover, an intelligent task scheduling algorithm was developed to check the availability of resources in the cloud, thereby reducing the waiting time. The experimental analysis of the presented algorithm shows that the performances like latency, waiting time, power consumption, etc., are minimized in this approach.

3. SYSTEM MODEL AND PROBLEM STATEMENT

IoT and CC infrastructure are integrated in smart healthcare to allow for real-time health monitoring. Sensors and IoT devices are used in IoT-based healthcare to gather patient data, such as blood pressure, cholesterol, body temperature, etc. These data can be used whenever needed because they have been moved to the cloud layer for storage. CC is the term used to describe the online distribution of computing services such as databases, software, storage, intelligence, etc. Allocating resources optimally is the main problem in CC. The RA in cloud infrastructure is displayed in Fig 1. Generally, in the cloud, the resources are allocated based on the user's request. However, the wide usage of the cloud network increases the load traffic, waiting time, latency, etc.

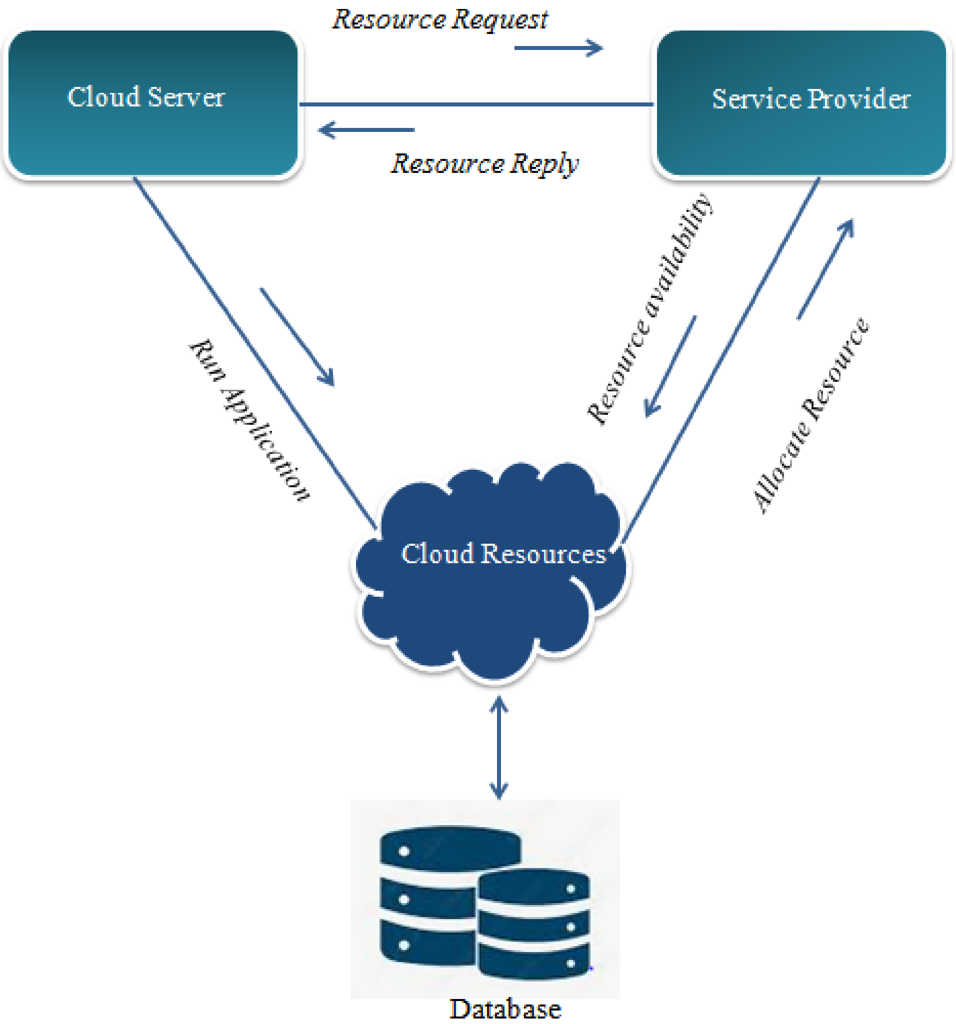

Figure 1. Resource Allocation in Cloud Environment

The RA in the cloud is of two major types namely: Dynamic and Static RA. But static RA is less effective because of high fluctuations in the cloud network. However, when compared with the static approach the dynamic method offers higher performance. Although different DRA methods are developed in the past, they face challenges in resource utilization, latency, and response time. Hence, an optimized neural-based RA method is established in this article to overcome the present challenges.

4. PROPOSED HPbFFNS FOR RESOURCE ALLOCATION

The use of medical sensors to gather patient health data from their locations is made possible by developments in smart healthcare. The cloud environment stores these IoT device data collection results for later processing. However, the heavy load traffic at the cloud servers causes failure in resource distribution, consumes more time, and increases the delay in the network. Hence, to resolve these issues in the IoT-cloud environment a novel hybrid Honey Pot-based Feed Forward Neural System (HPbFFNS) framework was developed in this paper to allocate resources to IoT healthcare applications in an optimal manner. This framework integrates the optimal features of Honey Pot Optimization [30], and Feed Forward Neural System [31]. In this framework, initially, the patient's medical data are collected using medical sensors in the IoT layer. These collected data from the IoT devices are forwarded to the IoT gateway layer for processing. In the gateway layer, the proposed HPbFFNS technique was incorporated to offer optimal load balancing.

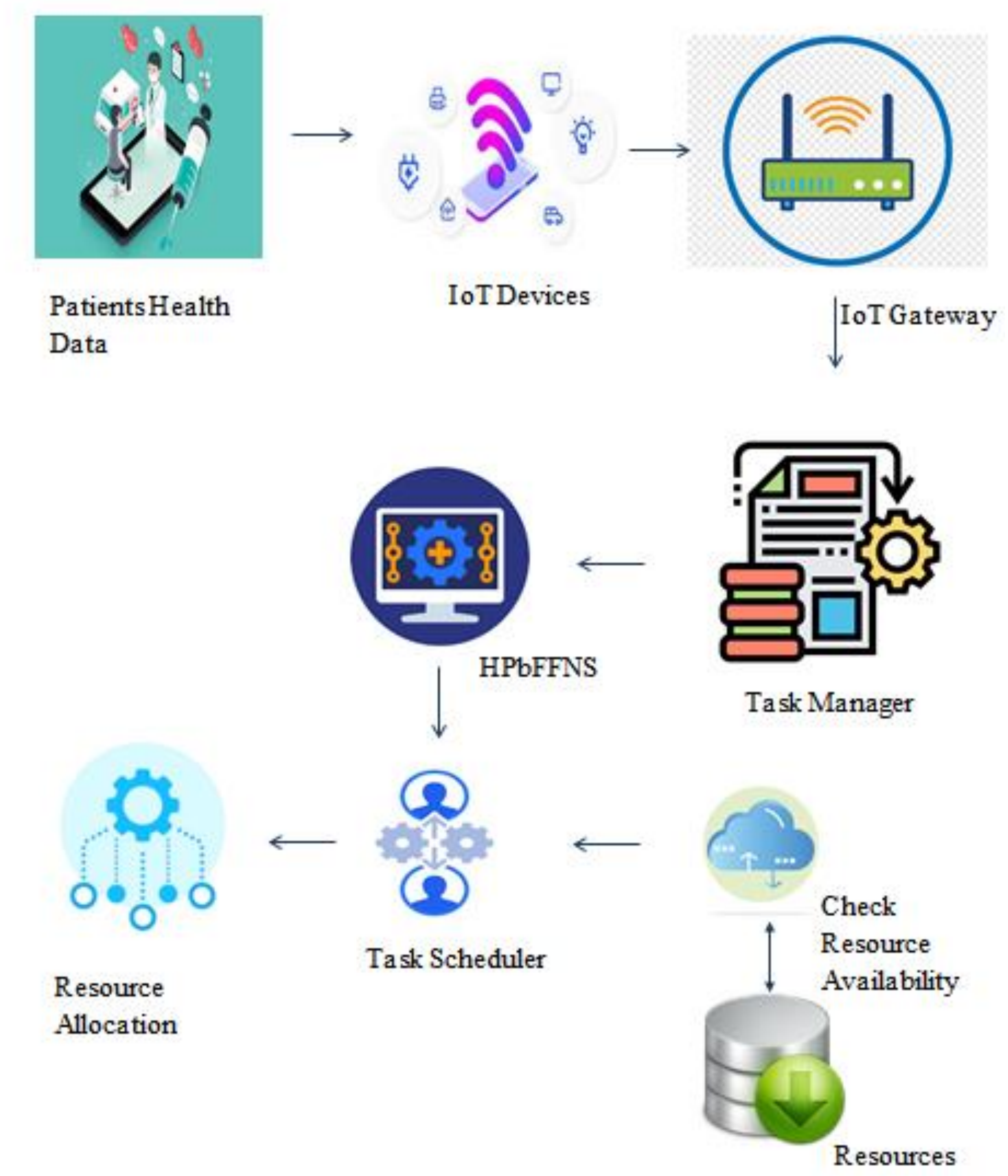

Figure 2. HPbFFNS Framework

In the gateway layer, there are three main components namely: Task manager, task schedular, and resource allocator. The task manager contains the view of all the processing tasks available in the network. It collects the task queues and forwards them to the scheduler. Moreover, it is the responsibility of the scheduler to check the availability of the resources in the cloud layer. Finally, the resource allocator assigns the resources as per the schedule to the healthcare applications considering the cost and energy consumption of each activity. The proposed framework is illustrated in Fig 2.

4.1. Data Collection

The integration of IoT in healthcare enables new scopes of patient healthcare through real-time monitoring and easy access to patients' health data. Moreover, it helps to facilitate healthcare progress virtually through telemedicine. The IoT devices utilized for gathering the patient's health information are a body temperature sensor, heart beat rate sensor, pulse-oximeter, fluid level sensor, etc. These sensors are placed inside the patient's body to provide real-time health monitoring. In addition, it eliminates the need for patients to travel to the healthcare units. In the cloud layer, the gathered information is stored for further processing. The IoT gateway is the hub that interconnects the IoT devices and sensors to the CC network. Moreover, it is responsible for filtering the collected data before transferring it out over the internet.

4.2. Task Scheduling

RA in an IoT-cloud environment involves three major steps namely: task management, RA and task scheduling. The task manager consists of the view of all incoming user requests (tasks) available in the cloud. It forwards the task queue to the task scheduler for RA. Here, the developed HPbFFNS model is integrated into the task schedular for reducing load traffic, waiting time, and latency. In the proposed model, the FFNN features are applied to the task scheduler to improve resource utilization. The FFNN is the simplest neural network, which can solve non-linear data faster. The problem-solving feature of FFNN is hybridized in the task scheduler to arrange the task effectively to improve the cloud performance. Initially, the scheduler checks the availability of the resources in the cloud. The task in the queue is expressed in Eqn. (1).

$$Q_s^{'} = [T_{s1}, T_{s2}, T_{s3}, T_{s4}, \dots . T_{sm}] \quad (1)$$

Where, $Q_s^{'}$indicates the task queue, T_s represents the task, and m denotes the number of the task in the queue. Then, the scheduler estimates the deadline of each task based on the start time of a task, and the number of VMs required to complete the task. The deadline for an individual task is estimated using Eqn. (2).

$$Dl_T(T_s) = \sum_{i=0}^{m} Vm_n(Ex_T - St_T) \quad (2)$$

Here, Dl_T denotes the deadline of the task, Vm_n indicates the number of VMs required to complete the task, Ex_T represents the execution time of the task, andSt_T defines the start time of the task. Then, the scheduler analyzes the priority, and length of each task to schedule the tasks optimally. The priority and task scheduling are represented in Eqn. (3) and Eqn. (4).

$$P_r^{'}(T_s) = L_{gt}\left(\frac{Dl_T - Sl_T}{Vm_n}\right) \quad (3)$$

$$S_t^{'}[T_s] = \{P_r^{'}(T_{si}) \geq P_r^{'}(T_{sm})\} \quad (4)$$

Where$P_r^{'}$ indicates the task priority, L_{gt} denotes the length of the task, and $S_t^{'}$represents the task scheduling.

4.3. Resource Allocation

Resource allocation is the method of assigning the available resources in the cloud to the users over the internet based on their demand. After scheduling, the available resources from the cloud are distributed to each task optimally. In the proposed framework, the parameters like resource availability, resource demand, and task completion time. Initially, the minimum available resources in the cloud must be determined to reduce the waiting time. The minimum resource availability is expressed in Eqn. (5).

$$M_R^* = \frac{T_R - m_{ax}(T_s)}{m_{ax}(T_s) - m_{in}(T_s)} \quad (5)$$

Where M_R^*determines the minimum resource availability, $T_R$refers to the total available resources in the cloud, $m_{ax}(T_s)$and$m_{in}(T_s)$ defines the maximum and minimum RA to the incoming task. The honey pot fitness solution for RA is expressed in Eqn. (6).

$$R_{SA} = Hp_f + T_{set} \times (M_R^*) + D_{mr} \quad (6)$$

Where R_{SA}defines the RA function, Hp_findicates the honey pot fitness, D_{mr}denotes the resource demand, and T_{set}refers to the appropriate time duration of the task.

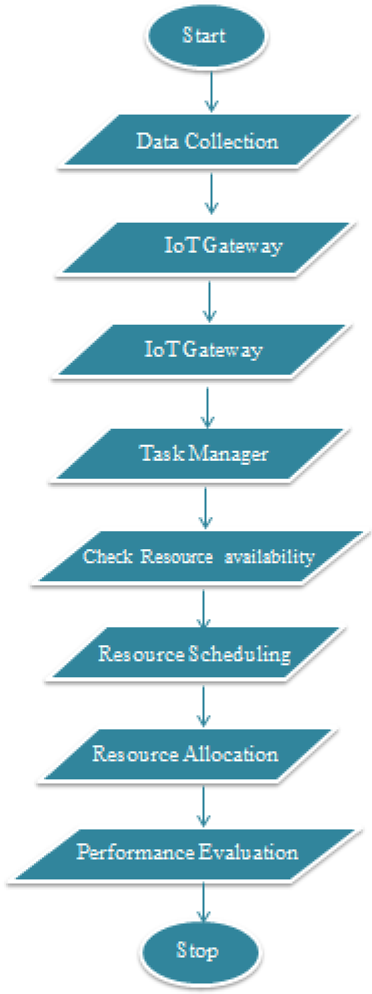

Figure 3. Flowchart of HPbFFNS

The honey pot fitness in the resource allocator enables it to analyze the resource demand and approximate time duration of the time. Thus, resource utilization is enhanced in the proposed method. The flowchart of the proposed model is illustrated in Fig 3.

5. RESULT AND DISCUSSION

A hybrid optimization-based RA framework was designed in this article to assign resources in the cloud environment for medical applications. Initially, the patient's health data was collected using IoT devices and sensors. The designed framework is implemented in MATLAB software, version R2020a.

5.1. Performance Analysis

To manifest the presented model performances, it is compared with existing techniques like Deadline-based Dynamic Resource Allocation (DbDRAA) [21], Improved Resource Allocation using Learning Classification Systems (IRA_LCS) [32], RA using Markov Learning Algorithm (RA_MLA) [27], and Task Scheduling based on Canonical PSO [33]. Moreover, the performance improvement percentage is also determined from the performance analysis.

5.1.1. Energy Consumption

Energy consumption defines the total energy consumed by the system to complete the task. For an efficient model, the energy consumption must be low to reduce the cost of the system. It is formulated in Eqn. (6).

$$E_{nc} = P_r \times \left(\frac{T_m}{1000}\right) \quad (6)$$

Where E_{nc} indicates the energy consumption, P_r denotes the applied power, and T_m refers to the time to complete the task.

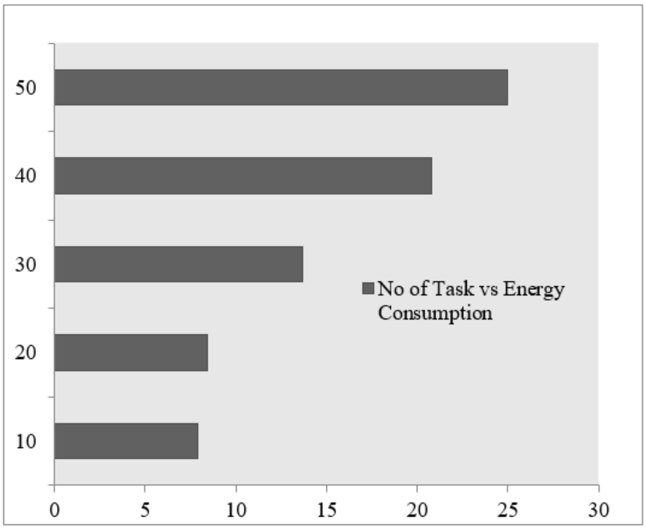

Figure 4. Energy Consumption Performance

In the presented work, the energy consumption is estimated by changing the number of tasks to 10, 20, 30, 40, and 50. When the task count increases, energy consumption increases in the system. It is observed that the energy consumption of the system for the assigned tasks is 7.89kWH, 8.4kWH, 13.7kWH, 20.8kWH, and 25kWH. The energy consumption performance analysis is shown in Fig 4.

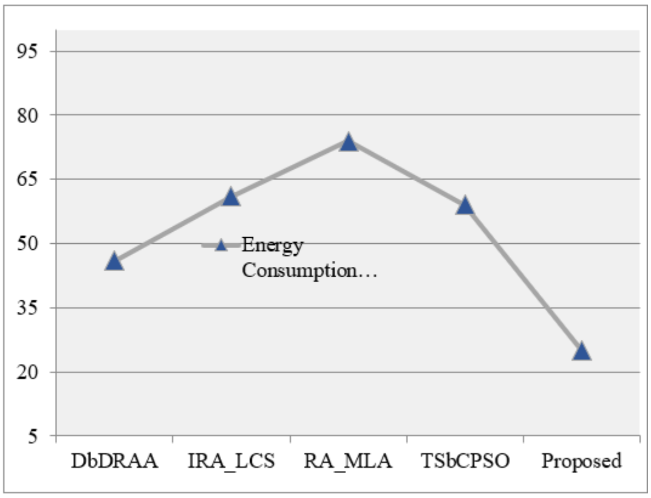

Figure 5. Comparison of Energy Consumption

Moreover, to manifest that the proposed model attained less energy to perform the task it is compared with some existing techniques like DbDRAA, IRA_LCS, RA_MLA, and TSbCPSO. The energy consumed by the traditional schemes is 46kWH, 61kWH, 74kWH, and 59kWH, respectively. This shows that the presented model consumed less energy to perform the tasks in the network. The energy consumption comparison is illustrated in Fig 5.

5.1.2. Response Time

Response time is the total time taken by the system to respond to the incoming request (task) for service. The service can be memory allocation, database query, etc. The response time is determined by adding the service and wait time to the network.

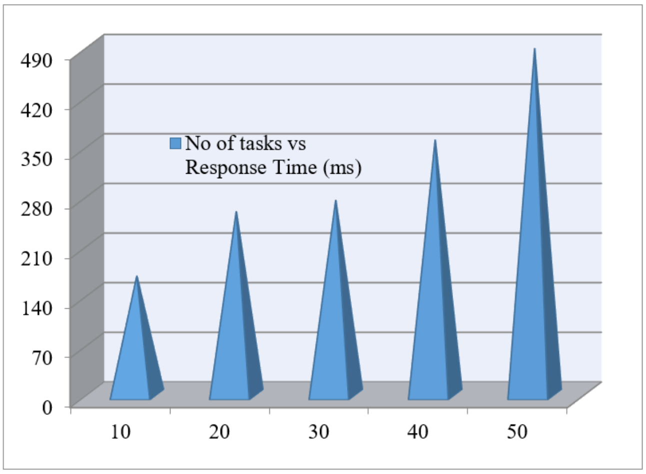

Figure 6. Response Time Analysis

Here, the response time of the presented system is determined by changing the number of tasks. The developed scheme achieved less response time of 168ms, 259ms, 275ms, 360ms, and 489ms, respectively for a different number of tasks (10, 20, 30, 40, and 50). The response time performance is illustrated in Fig 6.

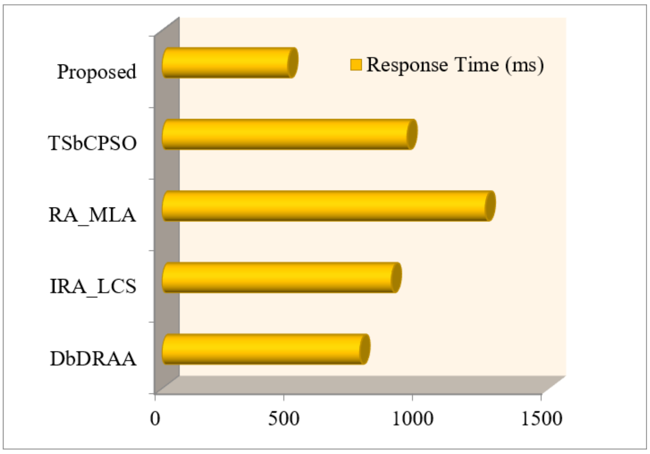

Figure 7. Comparison of Response Time

The response time attained by the existing algorithms is 768ms, 890ms, 1254ms, and 951ms. The response time comparison is shown in Fig 7. In addition, traditional frameworks like DbDRAA, IRA_LCS, RA_MLA, and TSbCPSO are executed on the same platform, and the response time is determined.

5.1.3. Resource Utilization

Resource utilization is one of the important parameters which determines cloud system performances. It defines the amount of resources used for the task. The presented algorithm earned a greater resource utilization rate of 96.7%. Further, it is compared with some existing techniques for validation purposes.

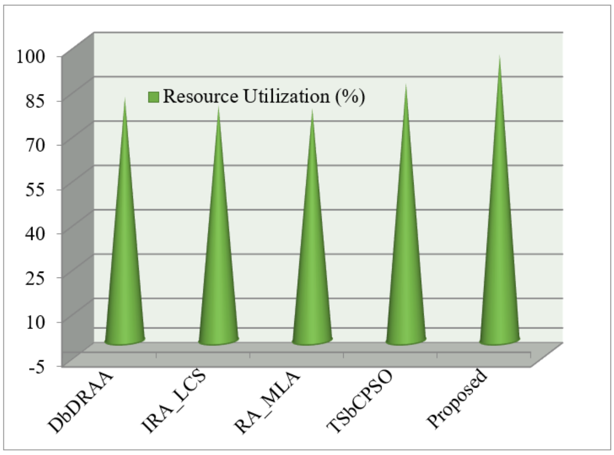

Figure 8. Comparison of Resource Utilization

The existing approaches like DbDRAA, IRA_LCS, RA_MLA, and TSbCPSO are executed in the same platform for medical applications. The resource utilization rate attained by the existing techniques is 82.3%, 79.46%, 78.6%, and 87%, respectively. The comparison of resource utilization rate is shown in Fig 8.

5.2. Discussion

In this article, an optimal dynamic resource allocation strategy was designed to allocate resources for IoT medical applications in the cloud infrastructure. It combines the honey pot optimization and feedback neural system to schedule and allocate resources optimally. The comparative performance of the developed scheme is tabulated in Table 1.

Table 1. Comparative Analysis

Techniques	***Response Time (ms)***	***Energy Consumption (kWH)***	***Resource Utilization (%)***
DbDRAA	768	46	82.3
IRA_LCS	890	61	79.46
RA_MLA	1254	74	78.6
TSbCPSO	951	59	87
Proposed	489	25	96.7

6. CONCLUSION

In recent times, one of the major concerns in the IoT-cloud infrastructure is the optimal resource allocation to incoming requests from different clients. To resolve this issue, an optimized neural-based DRA strategy was proposed in this article. The developed scheme involves three major steps namely: data collection, task scheduling, and RA. The FFNN attributes and honey pot fitness are integrated into the developed scheme to improve resource utilization by optimally assigning the available resources in the cloud. Furthermore, the performances of the accessible algorithm were estimated and validated with a comparative assessment. In addition, the performance improvement rate is determined from the comparative analysis. In the presented approach, the resource utilization is enhanced by 9.3%, the response time is minimized by 279ms, and the energy consumption is reduced by 21kWH. Thus, the designed model allocates resources optimally to the medical applications and improves the performances.

CONFLICTS OF INTEREST

Not applicable.

FUNDING STATEMENT

Not applicable.

ACKNOWLEDGEMENTS

The author would like to express his heartfelt gratitude to the supervisor for his guidance and unwavering support during this research for his guidance and support.

REFERENCES

[1] S. B. Sangeetha, R. Sabitha, B. Dhiyanesh, G. Kiruthiga, N. Yuvaraj, and R. A. Raja, “Resource management framework using deep neural networks in multi-cloud environment,” Operationalizing Multi-Cloud Environments: Technologies, Tools and Use Cases, pp. 89-104, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[2] M. Kandan, A. Krishnamurthy, S. A. M. Selvi, M. Y. Sikkandar, M. A. Aboamer, and T. Tamilvizhi, “Quasi oppositional Aquila optimizer-based task scheduling approach in an IoT enabled cloud environment,” The Journal of Supercomputing, vol. 78, no. 7, pp. 10176-10190, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[3] K. Peng, H. Huang, B. Zhao, A. Jolfaei, X. Xu, and M. Bilal, "Intelligent computation offloading and resource allocation in IIoT with end-edge-cloud computing using NSGA-III," IEEE Transactions on Network Science and Engineering, vol, 10, no. 5, pp. 3032-3046, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[4] R. Jeyaraj, A. Balasubramaniam, A. K. MA, N. Guizani, and A. Paul, "Resource management in cloud and cloud-influenced technologies for internet of things applications," ACM Computing Surveys, vol. 55, no. 12, 1-37, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[5] V. Jain, and B. Kumar, "Auction based cost-efficient resource allocation by utilizing blockchain in fog computing," Transactions on Emerging Telecommunications Technologies, vol. 33, no. 7, pp. e4469, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[6] Q. Li, P. Kumar, and M. Alazab, "IoT-assisted physical education training network virtualization and resource management using a deep reinforcement learning system," Complex & Intelligent Systems, pp. 1-14, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[7] Z. Wang, T. Lv, and Z. Chang, "Computation offloading and resource allocation based on distributed deep learning and software defined mobile edge computing," Computer Networks, vol. 205, pp. 108732, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[8] Z. Xiang, Y. Zheng, D. Wang, M. He, C. Zhang, and Z. Zheng, "Robust and cost-effective resource allocation for complex iot applications in edge-cloud collaboration," Mobile Networks and Applications, vol. 27, no. 4, pp. 1506-1519, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[9] Y. Kumar, S. Kaul, and Y. C. Hu, "Machine learning for energy-resource allocation, workflow scheduling and live migration in cloud computing: State-of-the-art survey," Sustainable Computing: Informatics and Systems, vol. 36, pp. 100780, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[10] N. Nithiyanandam, M. Rajesh, R. Sitharthan, D. Shanmuga Sundar, K. Vengatesan, and K. Madurakavi, "Optimization of performance and scalability measures across cloud based IoT applications with efficient scheduling approach," International Journal of Wireless Information Networks, vol. 29, no. 4, pp. 442-453, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[11] S. Masoudi, and F. Safi-Esfahani, "SM@ RMFFOG: sensor mining at resource management framework of fog computing," The Journal of Supercomputing, vol. 78, no. 17, pp. 19188-19227. 2022. [CrossRef] [Google Scholar] [Publisher Link]

[12] H. Chen, W. Qin, and L. Wang, Task partitioning and offloading in IoT cloud-edge collaborative computing framework: a survey. Journal of Cloud Computing, vol. 11, no. 1, pp. 86, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[13] S. Abedi, M. Ghobaei-Arani, E. Khorami, and M. Mojarad, "Dynamic resource allocation using improved firefly optimization algorithm in cloud environment," Applied Artificial Intelligence, vol. 36, no. 1, pp. 2055394, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[14] R. Rani, V. Kashyap, and M. Khurana, "Role of IoT-cloud ecosystem in smart cities: review and challenges," Materials Today: Proceedings, vol. 49, pp. 2994-2998, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[15] F. Firouzi, B. Farahani, and A. Marinšek, "The convergence and interplay of edge, fog, and cloud in the AI-driven Internet of Things (IoT)," Information Systems, vol. 107, pp. 101840, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[16] M. Prakash, V. Vijayaganth, F. D. Shadrach, R. Menaha, T. Daniya, and T. Guha, Improved Political Optimizer and Deep Neural Network-based Resource Management Strategy for fog Enabled Cloud Computing. In 2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon), pp. 1-6, 2022. IEEE. [CrossRef] [Google Scholar] [Publisher Link]

[17] S. Rahul, and V. Bhardwaj, "Optimization of Resource Scheduling and Allocation Algorithms," In 2022 Second International Conference on Interdisciplinary Cyber Physical Systems (ICPS), pp. 141-145, 2022. IEEE. [CrossRef] [Google Scholar] [Publisher Link]

[18] G. Vijayasekaran, and M. Duraipandian, "An efficient clustering and deep learning based resource scheduling for edge computing to integrate cloud-IoT," Wireless Personal Communications, vol. 124, no. 3, pp. 2029-2044, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[19] X. Sun, J. Jia, Z. Liu, Y. Li, B. Sun, and D. Liu, Resource Allocation and Load Balancing Based on Edge Computing in Industrial Networks. In 2022 IEEE International Conference on Smart Internet of Things (SmartIoT), pp. 250-251, 2022. IEEE. [CrossRef] [Google Scholar] [Publisher Link]

[20] R. Pradhan, A. K. Dash, and B. Jena, Resource management challenges in IoT based healthcare system. Smart Healthcare Analytics: State of the Art, pp. 31-41. 2022. [CrossRef] [Google Scholar] [Publisher Link]

[21] R. K. Naha, S. Garg, A. Chan, and S. K. Battula, "Deadline-based dynamic resource allocation and provisioning algorithms in fog-cloud environment," Future Generation Computer Systems, vol. 104, pp. 131-141, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[22] Z. Chang, L. Liu, X. Guo, and Q. Sheng, "Dynamic resource allocation and computation offloading for IoT fog computing system," IEEE Transactions on Industrial Informatics, vol. 17, no. 5, pp. 3348-3357, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[23] D. Baburao, T. Pavankumar, and C. S. R. Prabhu, "Load balancing in the fog nodes using particle swarm optimization-based enhanced dynamic resource allocation method," Applied Nanoscience, 13(2), 1045-1054, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[24] Y. Chen, Z. Liu, Y. Z hang, Y. Wu, X. Chen, and L. Zhao, "Deep reinforcement learning-based dynamic resource management for mobile edge computing in industrial internet of things," IEEE Transactions on Industrial Informatics, vol. 17, no. 7, pp. 4925-4934, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[25] S. Mishra, M. N. Sahoo, S. Bakshi, and J. J. Rodrigues, "Dynamic resource allocation in fog-cloud hybrid systems using multicriteria AHP techniques," IEEE Internet of Things Journal, vol. 7, no. 9, pp. 8993-9000, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[26] J. Praveenchandar, and A. Tamilarasi, "Retracted article: dynamic resource allocation with optimized task scheduling and improved power management in cloud computing," Journal of Ambient Intelligence and Humanized Computing, vol. 12, no. 3, pp. 4147-4159, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[27] S. M. Mirmohseni, C. Tang, and A. Javadpour, "Using Markov learning utilization model for resource allocation in cloud of thing network," Wireless Personal Communications, vol. 115, pp. 653-677, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[28] H. Liang, X. Zhang, X. Hong, Z. Zhang, M. Li, G. Hu, and F. Hou, "Reinforcement learning enabled dynamic resource allocation in the internet of vehicles," IEEE Transactions on

Industrial Informatics, vol. 17, no. 7, pp. 4957-4967, 2020. [CrossRef] [Google Scholar] [Publisher Link]
[29] W. Wei, R. Yang, H. Gu, W. Zhao, C. Chen, and S. Wan, "Multi-objective optimization for resource allocation in vehicular cloud computing networks," IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 12, pp. 25536-25545, 2021. [CrossRef] [Google Scholar] [Publisher Link]
[30] Á. Balogh, M. Érsok, L. Erdődi, A. Szarvák, E. Kail, and A. Bánáti, "Honeypot optimization based on CTF game," In 2022 IEEE 20th Jubilee World Symposium on Applied Machine Intelligence and Informatics (SAMI), pp. 000153-000158, 2022. IEEE. [CrossRef] [Google Scholar] [Publisher Link]
[31] J. Guo, C. K. Wen, and S. Jin, "Deep learning-based CSI feedback for beamforming in single-and multi-cell massive MIMO systems," IEEE Journal on Selected Areas in Communications, vol. 39, no. 7, pp.1872-1884, 2020. [CrossRef] [Google Scholar] [Publisher Link]
[32] M. Buvana, K. Loheswaran, K. Madhavi, S. Ponnusamy, A. Behura, and R. Jayavadivel, "Improved Resource management and utilization based on a fog-cloud computing system with IoT incorporated with Classifier systems," Microprocessors and Microsystems, 103815, 2021. [CrossRef] [Google Scholar] [Publisher Link]
[33] M. Z. Hasan, and H. Al -Rizzo, Task scheduling in Internet of Things cloud environment using a robust particle swarm optimization. Concurrency and Computation: Practice and Experience, vol. 32, no. 2, pp. e5442, 2020. [CrossRef] [Google Scholar] [Publisher Link]

AUTHORS

Kumarraja Andanapalli received his Ph.D. degree in Electrical Engineering from NIT Raipur in the year 2023. Received M.E & B.E Degree from SRKR Engineering College, Bhimavaram and ANITS, visakhapatnam from India in 2013 & 2008. Presently, working as an Assistant Professor in the EEE department in SRKR Engineering College, AP. His area of interest includes power system protection, Artificial Intelligence, Signal Processing Techniques, Internet of things.

Suresh Kumar M currently works as Associate Professor in the department of Computer Science and Engineering at Sri Venkateswara College of Engineering and has 17 years of Teaching experience. Suresh Kumar M completed his UG in the department of Information Technology at paavai engineering college, Namakkal and also completed his PG in the department of CSE with the specialisation of System Engineering and operational Research at Anna University in 2010. Also, I completed my doctorate in 2024 in Wsn. My primary research areas are data mining, cyber security, Deep learning and WSN. Also, I am working as an Assistant Placement officer in my college.

Arrived: 12.09.2024
Accepted: 15.10.2024