




Research Article

Cloud-Internet of Health Things (IOHT) Task Scheduling Using Hybrid Moth Flame Optimization with Deep Neural Network Algorithm for E Healthcare Systems

N. Arivazhagan,¹ K. Somasundaram ,² D. Vijendra Babu,³ M. Gomathy Nayagam,⁴ R. M. Bommi,⁵ Gouse Baig Mohammad,⁶ Puranam Revanth Kumar,⁷ Yuvaraj Natarajan,⁸ V. J. Arulkarthick,⁹ V. K. Shanmuganathan,¹⁰ K. Srihari ,¹¹ M. Ragul Vignesh,¹² and Venkatesa Prabhu Sundramurthy ¹³

¹Department of Computational Intelligence, SRM Institute of Science and Technology, SRM Nagar, Kattankulathur 603203, India

²Department of Computer Science Engineering, Chennai Institute of Technology, Chennai, Tamilnadu, India

³Department of Electronics and Communication Engineering, Aarupadai Veedu Institute of Technology, Vinayaka Mission's Research Foundation, Paiyanoor, Tamil Nadu, India

⁴Department of Computer Science Engineering, Ramco Institute of Technology, Rajapalayam, Tamilnadu, India

⁵Centre for System Design, Chennai Institute of Technology, Chennai, Tamilnadu, India

⁶Department of Computer Science Engineering, Vardhaman College of Engineering, Hyderabad, India

⁷Department of Electronics and Communication Engineering, IcfaiTech (Faculty of Science and Technology), Hyderabad, India

⁸Training and Research, ICT Academy, Chennai, Tamilnadu, India

⁹JCT College of Engineering and Technology, Coimbatore, Tamilnadu, India

¹⁰Department of Mechanical Engineering, J.N.N. Institute of Engineering, Kannigaipair, Tamilnadu, India

¹¹Department of Computer Science Engineering, SNS College of Technology, Coimbatore, Tamilnadu, India

¹²Department of Computer Science Engineering, Dhanalakshmi Srinivasan College of Engineering, Coimbatore, Tamilnadu, India

¹³Department of Chemical Engineering, Addis Ababa Science and Technology University, Addis Ababa, Ethiopia

Correspondence should be addressed to Venkatesa Prabhu Sundramurthy; venkatesa.prabhu@aastu.edu.et

Received 25 October 2021; Revised 8 November 2021; Accepted 30 November 2021; Published 5 January 2022

Academic Editor: M Pallikonda Rajasekaran

Copyright © 2022 N. Arivazhagan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Considering task dependencies, the balancing of the Internet of Health Things (IoHT) scheduling is considered important to reduce the make span rate. In this paper, we developed a smart model approach for the best task schedule of Hybrid Moth Flame Optimization (HMFO) for cloud computing integrated in the IoHT environment over e-healthcare systems. The HMFO guarantees uniform resource assignment and enhanced quality of services (QoS). The model is trained with the Google cluster dataset such that it learns the instances of how a job is scheduled in cloud and the trained HMFO model is used to schedule the jobs in real time. The simulation is conducted on a CloudSim environment to test the scheduling efficacy of the model in hybrid cloud environment. The parameters used by this method for the performance assessment include the use of resources, response time, and energy utilization. In terms of response time, average run time, and lower costs, the hybrid HMFO approach has offered increased response rate with reduced cost and run time than other methods.

1. Introduction

Networking devices at various levels of the network architecture deploy processing and storage capabilities in fog computing. Data centers (DCs) in cloud have schedulers, and the schedulers decide where to deliver requests. Such a choice is frequently dependent on a number of factors, including the requests themselves as well as the cloud resources. The result is a reduction in latency and increased proximity of services and computations to end users.

Data are getting increasingly voluminous as the number of internet-connected devices grows and the Internet of Things (IoT) takes off. According to Cisco predictions, there will be 50 billion Internet-connected gadgets by 2020 [1–4]. Large volumes of heterogeneous data, exchanged rapidly, and, at high speeds, will be generated and exchanged by these devices, which is known as “Big Data.” IoT applications create a lot of data that is latency sensitive, which means they need a processing response in real time [5].

In cloud computing environment, Internet of Health Things (IoHT) is permitted to use cloud interactions. It enables mobile users to achieve attractive characteristics which make it marketable. In cloud, interaction between the IoHT devices can form a new environment, which is referred as cloud-IoHT system [6–10].

Some IoHT devices generate data that expire in seconds (the data expiry time) following the generating time. Because of the time-sensitive nature of many applications, processing this data in the cloud could become a performance issue. In a fog and cloud computing architecture, each component completes what the other is missing. It is hoped that the connection between fog and cloud will enable low-latency services for billions of IoT devices and applications in the future. IoT devices generate enormous amounts of data, which this tool greatly aids in monitoring and controlling.

The jobs to transmit the data from IoHT to cloud servers offer a variety of computational and storage requirements that must be met in order to be accomplished. Offloading IoHT duties to cloud resources requires consideration of various quality of service (QoS). Not only from the perspective of IoHT applications for end users but also from the perspective of the system designer. The optimization problem of task scheduling pertains to the process of identifying optimal fog or cloud resources for allocating IoHT jobs under specified constraints. There are several objectives for optimising QoS parameters for both end users and system designers in designing a scheduler for dynamic incoming IoT workloads. An integer linear programming (ILP) [11–15] of NP-hard complexity can be used to formulate the job scheduling optimization issue in this context. It should be highlighted that conventional mathematical methods cannot discover an optimal solution in polynomial processing time for such issues [16–20].

When faced with NP-hard issues such as task scheduling, population-based metaheuristic algorithms have been shown to be successful optimization strategies. When applying these techniques, a random sample of the population is created using a uniform distribution, and it is hoped that, over time, the population will converge to an optimal

solution. In order to identify the best solution, the search process should be divided into two phases: exploration and exploitation. Starting with a varied population will allow us to investigate all options in the search space and identify areas with high potential. The study should then be able to leverage the potential areas in search of a worldwide solution that works perfectly. The shift from exploration to exploitation, on the other hand, is not always smooth, especially if the population becomes caught in the local optima too early and premature convergence occurs.

For a long time, academics working on various heterogeneous computing systems have debated how to best schedule tasks. In the context of fog computing, where exponential IoT jobs are generated dynamically, there are no acceptable specific solutions to the task scheduling problem [21]. Due to the distinct restrictions of IoT activities and fog resources, the search space for the IoT and cloud environment is really more complex than traditional computing systems. IoT jobs are not always tolerant of delays, despite the fact that task scheduling is notoriously difficult. So, the task scheduling architecture should allow IoT applications to be handled quickly while avoiding the complexity of the IoT infrastructure [22–26].

This study provides a smart model approach for the best task schedule of hybrid moth flame optimization (HMFO) for cloud computing integrated in the IoHT environment over e-healthcare systems. The HMFO guarantees uniform resource assignment and enhanced QoS. The parameters used by this method for the performance assessment include the use of resources, response time, and energy utilization.

The outline of the paper is given below. Section 2 provides the problem formulation. Section 3 discusses the task scheduling in cloud using HMFO. Section 4 evaluates the entire work in terms of various performance metrics. Section 5 concludes the entire work with possible directions of future scope.

2. Problem Formulation

The problem formulation [25] is defined with the following assumptions that includes past computational time over each executing task and the knowledge of computational overhead for scheduled task on the IoHT-cloud model in e-healthcare systems.

These two constraints are modelled as in equations (1)–(5):

$$\gamma_i = \frac{c_i}{p_i}, i = 1, 2, \dots, N, \quad (1)$$

$$r_{ij}(\forall i) = \begin{cases} 0, & j = 1, \\ d_{ij-1}, & j = 2, 3, \dots, n_i, \end{cases} \quad (2)$$

$$d_{ij} = r_{ij} + p_{ij}, \quad i = 1, 2, \dots, N, \quad j = 1, 2, \dots, n_i, \quad (3)$$

$$s_{ij} < s_{i'j'} < f_{ij}, \quad (4)$$

$$\text{if } p(\tau_{i'}) > p(\tau_i) \&\& \tau_{i'j'} \geq r_{ij} \forall i, j, i', j', \quad (5)$$

where i' is the task index, j' is the executed index of j^{th} task, p is the priority, N is the total number of tasks, T is the total time scheduled for carrying out the task, n_i is the total number of tasks executed, r_{ij} is the total release time (j) for a specific task τ_i , s_{ij} is the start time (j) for a task execution τ_i , f_{ij} is the finish time (j) for a task execution τ_i , and i is the execution time τ_i .

3. Task Scheduling Using HMFO

A multiobjective optimization is considered as a conflicting objective problem required to be optimized as defined in the following equation:

$$\text{Min } F(x) = (f_1(x), f_2(x), \dots, f_m(x))^T, \quad (6)$$

where $x \in X$ is regarded as the decision space.

The objective function conflicts with each other. The Pareto dominance performs an evaluation between the obtained solutions. With $v \in X$, the sample u dominates v iff, equation (7) denotes the relationship:

$$\forall i: f_i(u) \leq f_i(v) \wedge \exists j: f_j(u) < f_j(v). \quad (7)$$

If a solution x^* fails to dominate the other, the solution is treated as Pareto optimal one.

Figure 1 shows the IoHT-cloud architecture for sensing the environment, data acquisition, and data transmission from the source IoHT node to the destination cloud server. Three different levels are used in IoHT-cloud e-healthcare system architecture.

To address this issue, the suggested architecture makes use of a control plane composed of an MFO model integrated with a GA algorithm. So much responsibility is placed on the deep sense model that it ensures the connection or path is not overburdened with data packets. Heavy traffic can be checked with optimum packet selection of routing paths via optimum packet delivery in specific circumstances.

Massive amounts of IoHT data are subsequently transferred in packets to the destination node using the data plane. The integration of an MFO [13] model with the GA model aligns the purpose of the current model to improve cloud scheduling, which governs cloud routing.

Variations in VM computing power can be found in cloud data centers, meaning that even when performing equivalent activities, VMs can perform significantly differently. Because workloads vary, different VMs have variable levels of performance efficiency, allowing tasks to be distributed across them. Virtual machines with superior performance capacity would be given additional jobs. The aforementioned reason causes certain VMs to be overloaded while the rest are inactive. This would be a waste of time and money in the long run.

An unbalanced load on the virtual machines reduces the cloud DC efficiency. Consequently, the goal of this research is to better utilise resources under average load conditions for VMs, according to the upgraded HMFO.

3.1. HMFO. This section provides the scheduling details that combine HMFO [26] (Figure 1) with deep layers of neural

network (DNN). The weight may influence the task selection process; those who are unable to deal with the problem will quickly be excluded. This is going to damage decent populations. New people delivered by crossover and mutation operations are very less. Although reductions in specific weighing can increase likelihood of looking at the global ideal, their mounting rate is lower.

The optimization of swarm molecules can be applied to the genetic algorithm in order to maintain great performance. DNN and HMFO are supporting the hybrid algorithm. The DNN can be used for a global investigation within the underlying time frame. The HMFO can be connected with the speed of the mix rate over the later period due to the reduced population size. In the latter years, after usual cycles, people will thus become closer to the international ideal arrangement at moderately high assembly rates.

The framework is similar to other population-based algorithms, where the initial step involves the generation of random solution. The initial solution is a candidate solution for the given scheduling problem in cloud space, and it is assessed via a fitness function. The MFO combines the fitness solution in order to fit the requirement of job scheduling, and this is iterated in repeated intervals to obtain optimal solution.

3.1.1. Fitness Function. Fitness is used to measure the level of intelligence of the population which could almost certainly achieve or help to find the ideal solution. His choice is therefore very important. The work process plans to achieve lower flow times and efficiency. Here, DNN is employed for finding the fitness of HMFO (Algorithm 1).

Multilayer perceptron (MLP) is an another form of feedforward neural network architecture with three different layers including input, output, and hidden (Figure 2(a)). The MLP designed with a single hidden layer using the approximation theorem estimates the output with the reduced error rate.

There are dropouts fitted to the DNN (Figure 2(b)) to reduce data overfitting, and the neurons are eliminated throughout training. As a result, the selected error function (using cross entropy) reduces the error during DNN training as

$$E = \sum_{i=1}^n y \log o_N + (1 - y) \log (1 - o_N). \quad (8)$$

The MLP tends to vary its weights based on the error function at the time of training and this reduces the probable error. The input dataset size D , for the classification of MLP, where the probability of errors $P(y|x)$ at the output is influenced by the hidden layers.

3.1.2. Crossover. Many characteristics such as average fitness and diversity change in the evolutionary process. Therefore, the entire algorithm programming process cannot be adapted to a fixed parameter. This algorithm is used for the dynamic transverse operators. The process of early selection

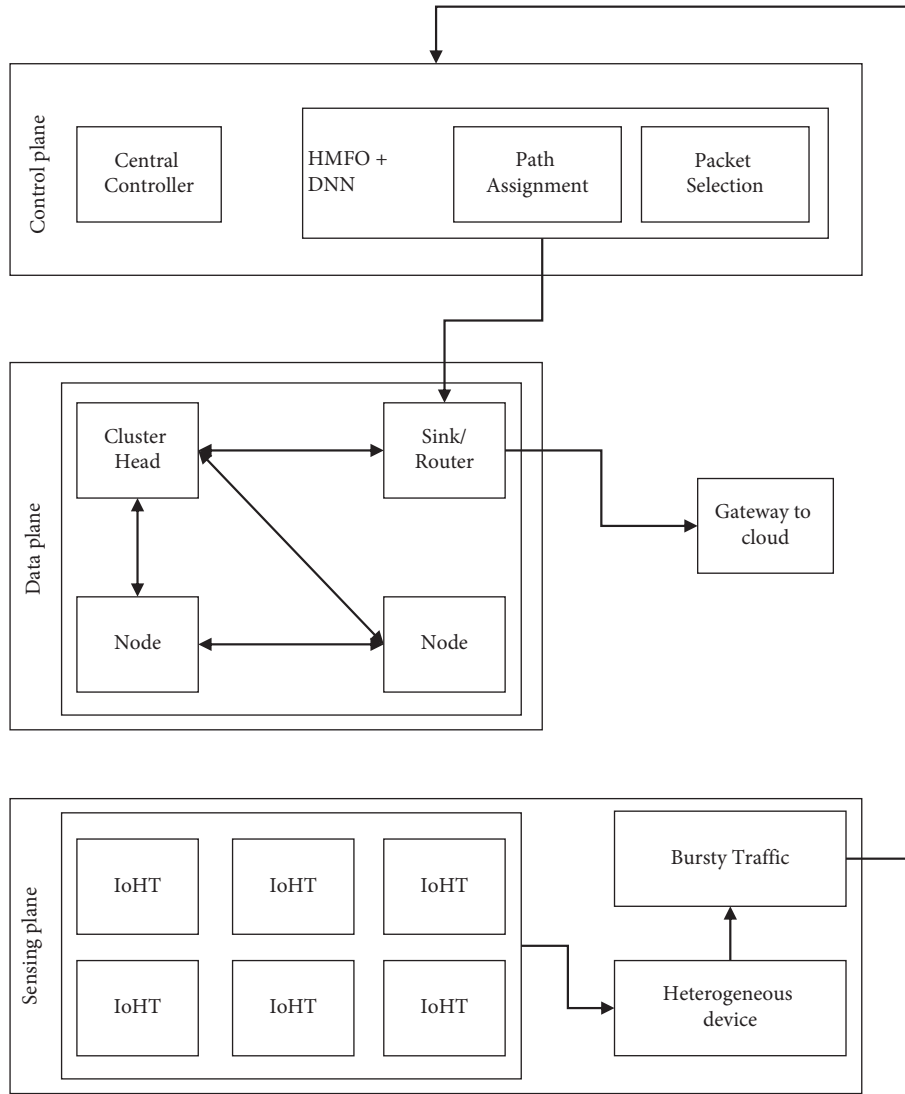


FIGURE 1: Architecture for time overhead minimization using hybrid HMFO.

```

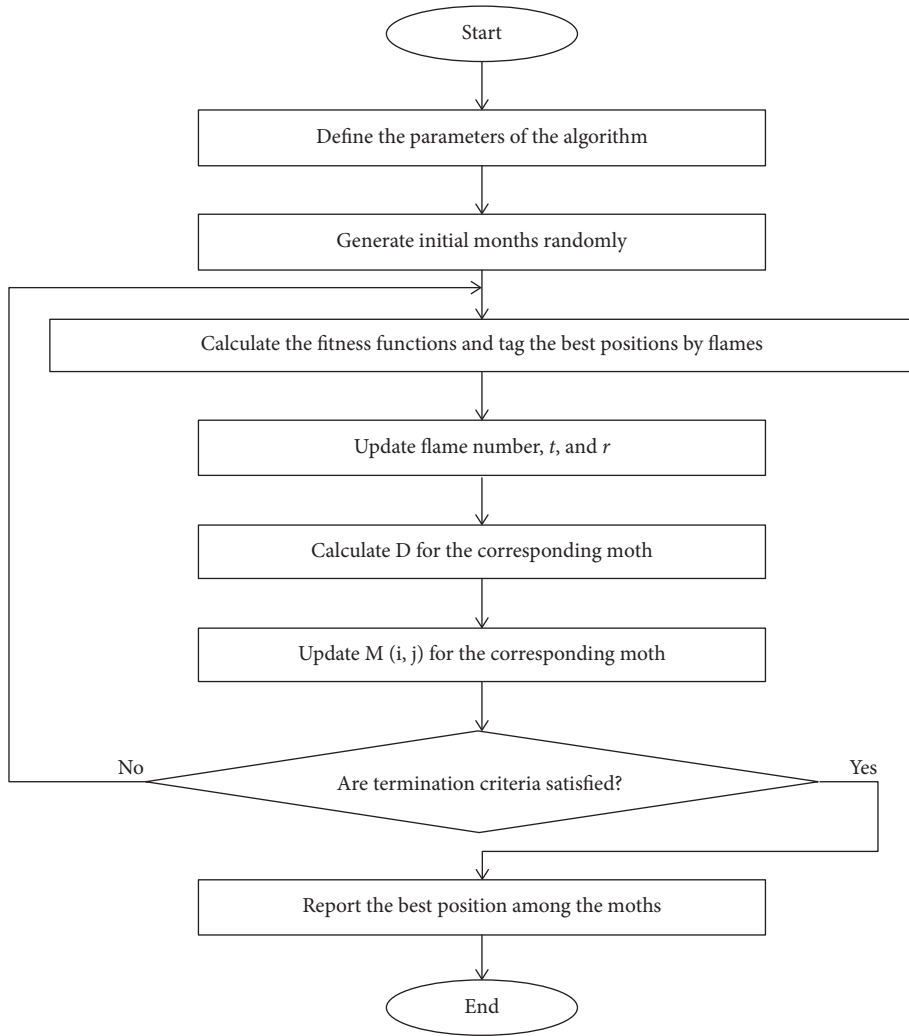
For  $i = 1 : N$ 
  For  $j = 1 : N$ 
     $M(i, j) = (ub(i) - lb(i)) * \text{rand}() + lb(i);$ 
  End
End
OM = FitnessFunction ( $M$ );

```

ALGORITHM 1: Generation of initial solution using MFO.

damages the diversity of the population. Therefore, new and optimal people must be searched hard. In the early pc period, the likelihood of crossover is quite large. However, people concentrated primarily on areas in the process of later evolution near the optimal solution zone.

In order to reduce p_c , certain well discovered genes are now maintained. This algorithm sets a threshold t , which is applied in the crossover process to calculate the average progeny fitness. In addition, the fitness of populations is used for minimum average.



(a)

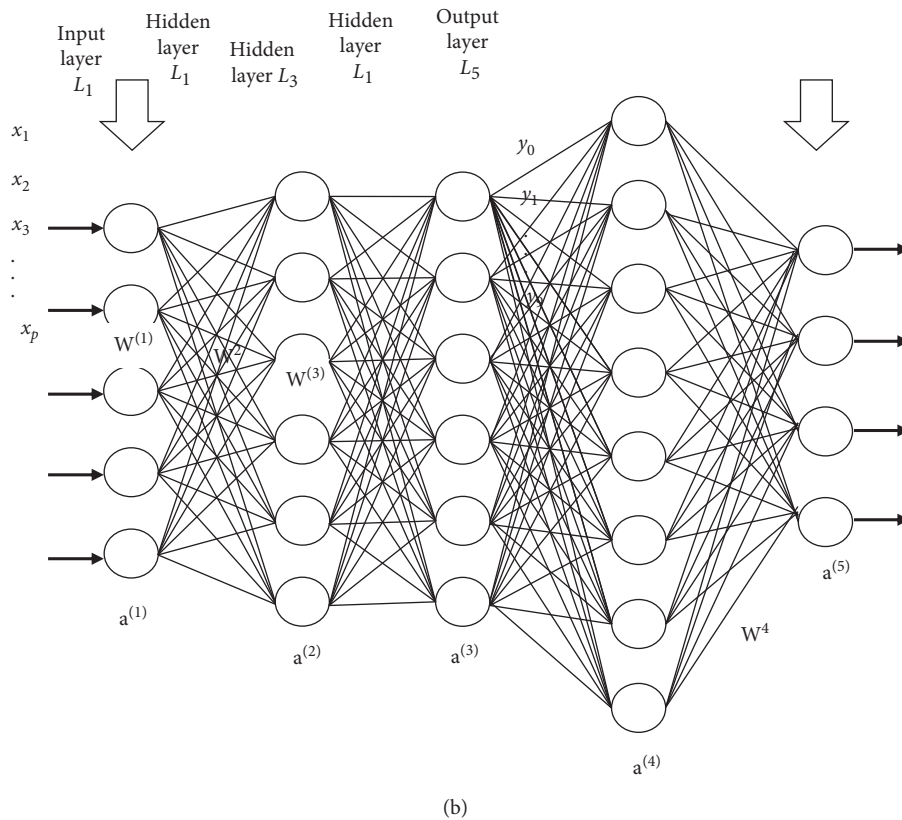


FIGURE 2: (a) Moth flame optimization. (b) DNN architecture.

The average fitness in procreation populations will be below the minimum average fitness if it is maintained over several generations. Then, the value of p_c lies between 90% of p_c and minimum probability of crossover.

3.1.3. Mutation. This paper adopts dynamic transmission operators, such as crossover operators. Instead of convergent algorithms in the population, the probability of mutation p_m can decrease according to rich genetic features. However, if the algorithm is close to or converged, p_m may be extended according to the population rate constantly or slightly changed. This algorithm sets a threshold for the calculation of the mutation rate. Furthermore, the minimum progress rate is determined for populations. If the evolution rate of the population for t iterations is continued, the evolution gets lowered than the minimum threshold limit. The value is regularly updated as a lesser value with the maximum probability of mutation.

3.1.4. Scheduling Using MFO. A task response time can be defined as the time it takes from submission to receipt of an answer in VM. VM responsiveness can be improved by balancing the load across all of their resources, including the CPU and RAM. A simple approach to reducing machining time and reaction time requires transferring work from one overburdened VM to another, which is known as the load-

balancing procedure here. It is critical that the VM communicates its load-balancing capacity in order to keep things running smoothly.

Computer nodes, storage nodes, and schedulers will all be hosted on management nodes. The scheduler selects appropriate computing nodes for the VMs based on the demands, and it keeps track of requests received to each one. In order to store the fingerprints associated to the running VMs, data centers are integrated. In a running VM, the task of the scheduler is to create a map of the VM with the IoHT node. The administration nodes are used to save the fingerprints. The fingerprints would be adequately removed from the management nodes once the process of the VM meeting terminates.

3.1.5. Load Balancing. There are two steps to the load balance strategy, according to this research. To begin with, the study employs a task planning technique that takes into account the varying needs of each user while still maximising resource use. In order to optimise the overall Cloud-IoHT performance, load balancing is utilised to map IoHT tasks to full VMs, followed initially by the VMs in resource hosting tasks.

Certain investigations, such as the identification of CPU uses and the assessment of memory requirements, must be accomplished in the first phase itself with a determination of the number of iterations in the first phase. The next step is to

```

Initialize the population using Moths
Initialize IoHT
Initialize VM
Assign the threshold value
While VMs are balanced with loads
if (load in VMs < threshold)
{
  do
  {
    Assign a task to VM
    Sort the task
    Estimate the moth distance
    Update the moth distance
  } while (load in VMs < threshold)
}
else
Update the new solution onto the moths
Search similar neighborhood VM
Check if the load is balanced or not in VMs

```

PSEUDOCODE 1: VM ware analysis.

determine the resources that are currently available and the resources. Based on these resource requirements, instances can be deleted or added, and the final prescribed status is determined.

The proposed technique wisely performs load checks on VMs on a regular basis, and based on such checks, the following strategy is utilised to determine load migration. Identifying the loading conditions of VMs is the key purpose of this algorithm, which differentiates between loaded VMs and loaded VMs based on the differentiation of machines. Once this is done, loads are transferred from loaded VM to loaded VM.

As a result of this procedure, the loads are dispersed equally. As a result, reaction time is cut in half, and resource efficiency is improved, thanks to more evenly distributed load. In a way, this duty of observation is similar to the moth search for food. This task threshold value is still being approached. A task minimum loaded VM is determined by an optimization that stops when the threshold is achieved (Pseudocode 1).

4. Results Analysis

In this section, the simulation is conducted to test the efficacy of the proposed model on job scheduling. The HMFO optimization is verified using performance measures including payload routing, delay, packet delivery ratio, and network lifetime. The study uses Google Cluster Dataset (<https://github.com/google/cluster-data>), where it is used to train the HMFO for future scheduling of jobs. The parameters are chosen from the Google cluster dataset that decides the number of VMs, cloud server, and the hardware specifications. The entire system is implemented on a

CloudSim simulator. The simulation is validated in terms of cost and delay and payload. The metaheuristic optimization is compared with conventional models such as bees swarm optimization (BSO) and ant colony optimization (ACO).

4.1. Simulation Results and Discussion. Figure 3 depicts the PDR, which includes IoHT source node packet transmission and sink node packet reception. According to the findings, when pause times are shortened, the PDR decreases significantly. The PDR, on the contrary, decreases as the number of session increases, and it is already higher than that of a traditional ANN.

The improved performance in the deep learning model is attributable to the sink node improved ability to calculate data transmission patterns. When data rates generated on IoHT devices do not match, typical systems do not calculate alternate pathways. As a result, the HMFO model performance is deemed stable, and the routing connection is strengthened by improved route stability and decreased link failure.

Figure 4 depicts the EAD results as the pause time increases. In comparison to other methods, such as ANN and reinforcement education, the EAD is significantly smaller. The choosing of lengthier routes may contribute to network congestion if there is a growing delay.

Metaheuristic algorithm learning, on the contrary, has no effect on distributing the workload. When it comes to computing power, a smaller data transmission rate is better because it enhances the model computational capability over time. The NRP and NMP show results at various sessions, where with increasing sessions, the results are found to get lowered as in Figures 5 and 6.

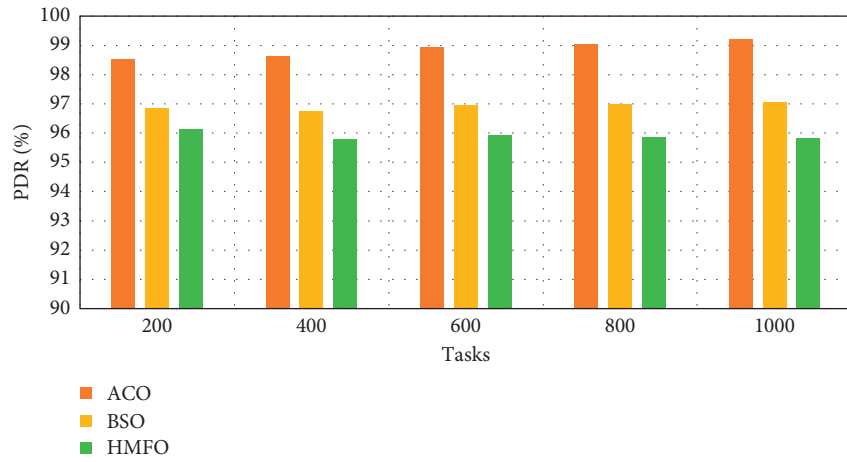


FIGURE 3: Packet delivery rate.

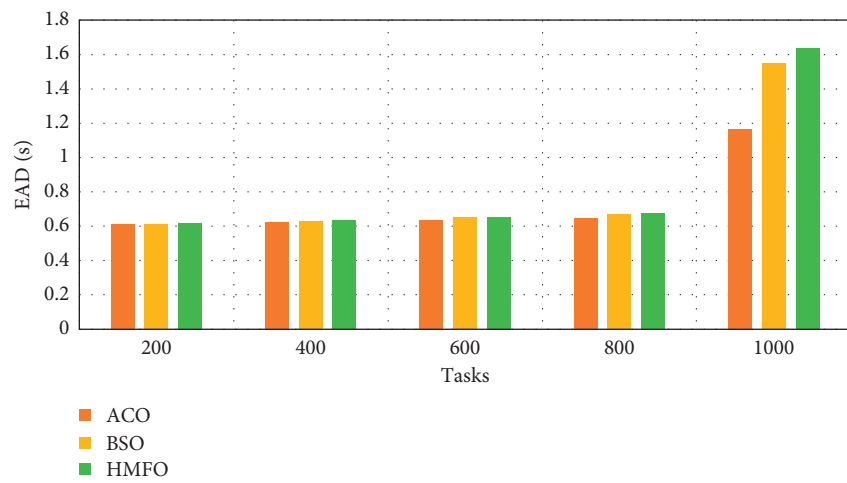


FIGURE 4: Average end-to-end delay.

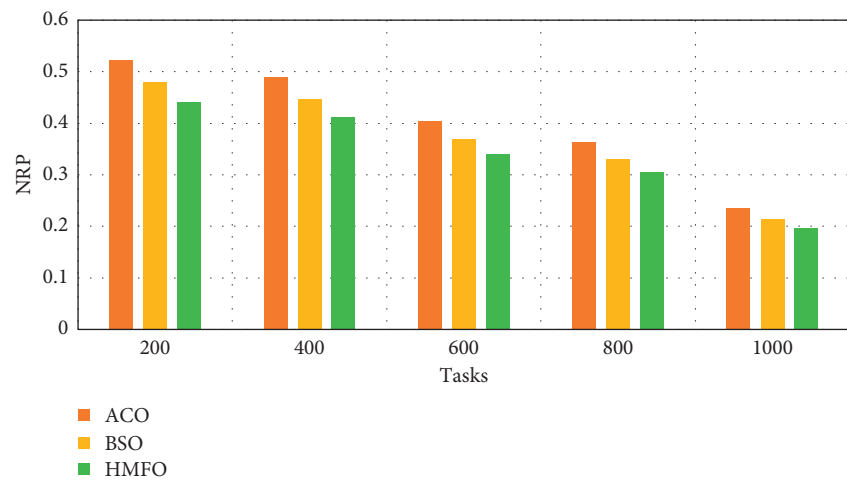


FIGURE 5: Normalized routing payload.

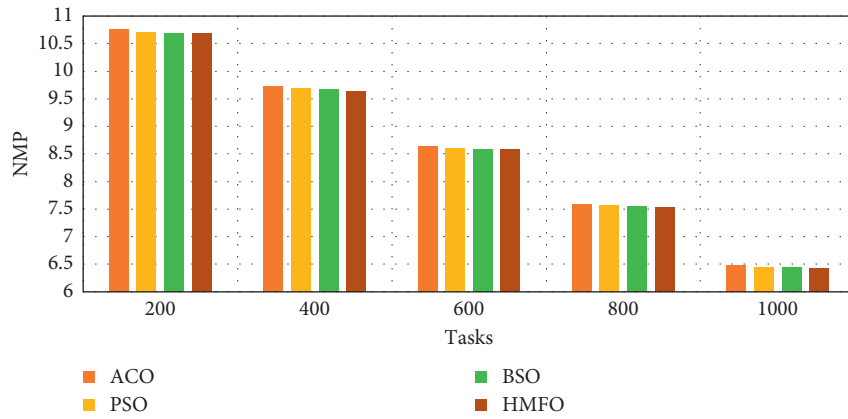


FIGURE 6: Normalized MAC payload.

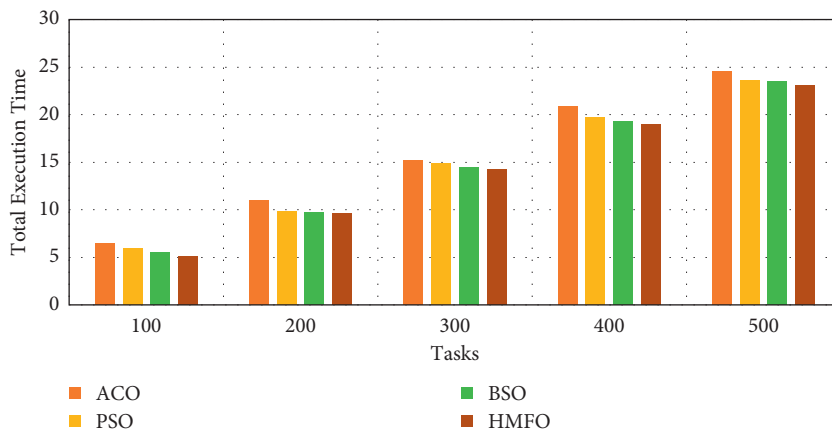


FIGURE 7: Makespan for different numbers of tasks.

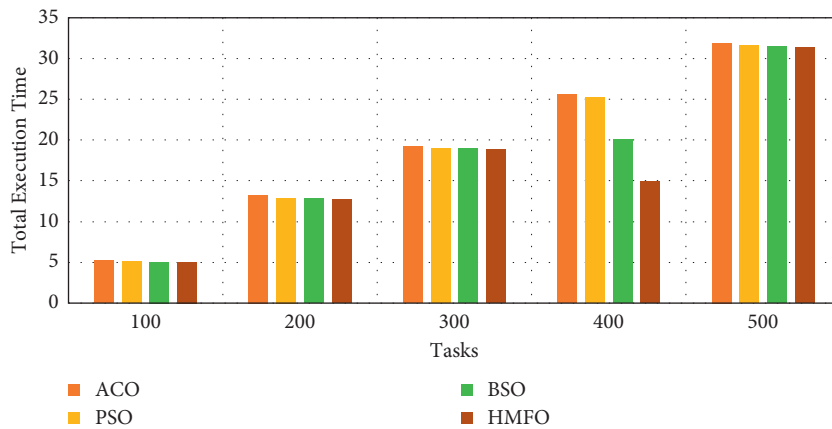


FIGURE 8: Total execution time.

Figure 7 depicts the average total timeline for various scheduling approaches for various jobs. In comparison to the HMFO strategy, the HMFO strategy uses less make up. HMFO output has dipped by 13% from its peak. The primary reason for this is that the planned strategy takes on the characteristics of the tasks, resulting in a total implementation time of the tasks' characteristics.

While there are only a few activities to complete, there are plenty of materials available (Figure 8). When comparing the HMFO technique, the execution time is extremely long. As more of the search region is uncovered, the HMFO algorithm shifts into a better position. When compared to the conventional technique, HMFO saves 8 to 16% of the time. Many other measures show this to be the case, such as

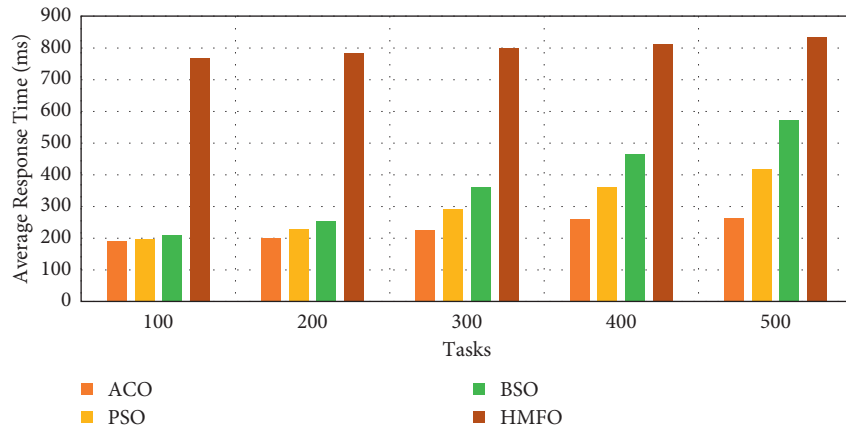


FIGURE 9: Response time.

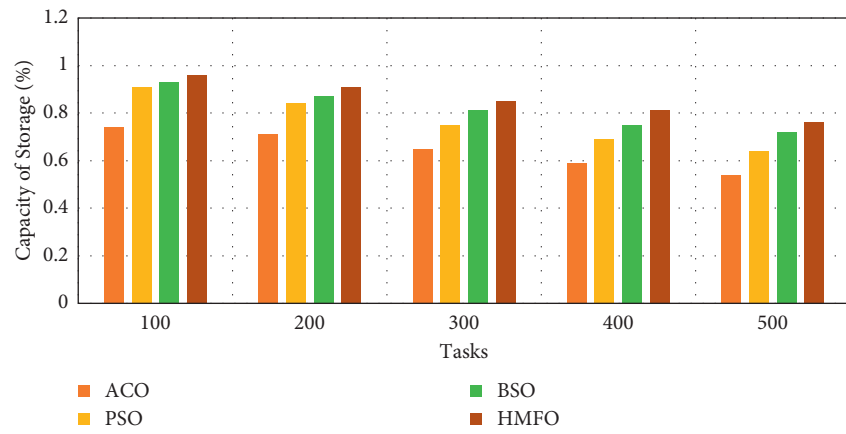


FIGURE 10: Storage capacity (%).

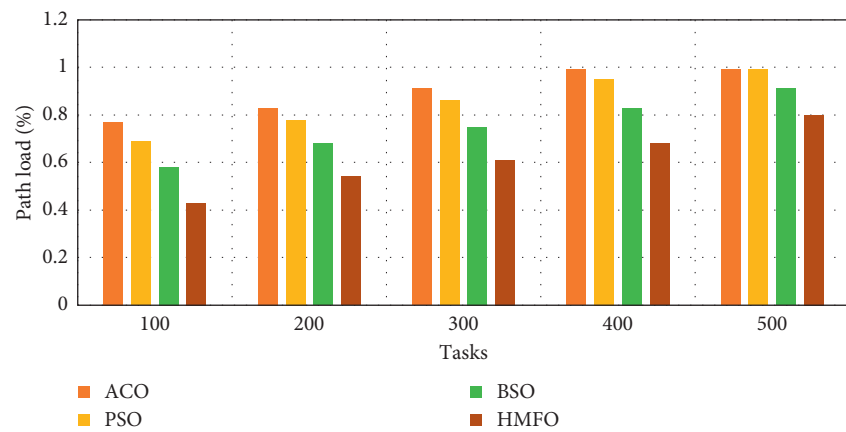


FIGURE 11: Path load (%).

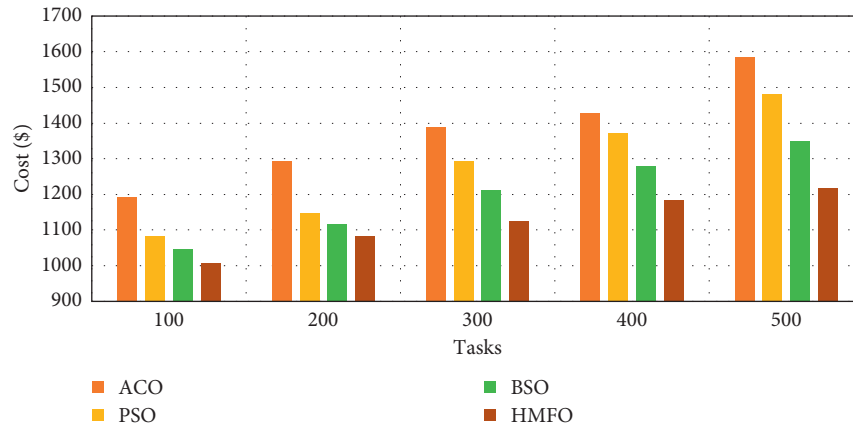


FIGURE 12: Cost (\$).

enhanced average reaction time (Figure 9), higher storage capacity (Figure 10), reduced path load (Figure 11), and decreased cost (Figure 12).

5. Conclusion

With HMFO, the routing performance in IoHT-Cloud is improved while the stability of the routing is increased. HMFO efficiently routes packets by adjusting to the data acquisition pace of IoHT nodes. In order to capture and transmit data without design flaws, IoHT-Cloud with e-healthcare system architecture is successfully controlled. Simulated results show that when EAD decreases, PDR increases as stability improves, and the reverse is also true. In contrast, HMFO minimises PDR on long routes, which biases the samples and assures that the system scales equally well. For shorter routes, the HMFO routing model scalability is excellent. The HMFO performance on high-speed packet routing provides reliable and increasing rate of packet transmission thus increasing IoHT-Cloud longevity in e-healthcare system architecture.

In future, the study may offer the mobile users with improved flexibility and reduced deployment costs using the deep learning training model. The cloud-IoHT technology amended in e-healthcare systems suffers mainly from energy balancing issues, and this should be addressed prominently on IoHT sensors.

Data Availability

The datasets used and/or analyzed during the current study are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] S. M. Nagarajan, G. G. Deverajan, P. Chatterjee, W. Alnumay, and U. Ghosh, "Effective task scheduling algorithm with deep learning for Internet of Health Things (IoHT) in sustainable smart cities," *Sustainable Cities and Society*, vol. 71, Article ID 102945, 2021.
- [2] P. Wang, K. Li, B. Xiao, and K. Li, "Multi-objective optimization for joint task offloading, power assignment, and resource allocation in mobile edge computing," *IEEE Internet of Things Journal*, vol. 19, no. 2, p. 1, 2021.
- [3] M. Mukherjee, V. Kumar, D. Maity et al., "Delay-sensitive and priority-aware task offloading for edge computing-assisted healthcare services," in *Proceedings of the GLOBECOM 2020-2020 IEEE Global Communications Conference*, pp. 1–5, IEEE, Taipei, Taiwan, December 2020.
- [4] S. Ketu and P. K. Mishra, "Internet of healthcare things: a contemporary survey," *Journal of Network and Computer Applications*, vol. 2021, Article ID 103179, 2021.
- [5] A. Lakhani, Q.-U.-A. Mastoi, M. A. Dootio et al., "Hybrid workload enabled and secure healthcare monitoring sensing framework in distributed fog-cloud network," *Electronics*, vol. 10, no. 16, p. 1974, 2021.
- [6] R. Chudhary and S. Sharma, "Fog-cloud assisted framework for heterogeneous internet of healthcare things," *Procedia Computer Science*, vol. 184, pp. 194–201, 2021.
- [7] V. Prabhu, V. Parthasarathy, U. Thirunavukkarasu, A. M. Judith, and S. Jagadeesan, "An energy-efficient centralized dynamic time scheduling for internet of healthcare things," *Measurement*, vol. 186, Article ID 110230, 2021.
- [8] L. Ning, Y. Ali, H. Ke, S. Nazir, and Z. Huanli, "A hybrid MCDM approach of selecting lightweight cryptographic cipher based on ISO and NIST lightweight cryptography security requirements for internet of health things," *IEEE Access*, vol. 8, pp. 220165–220187, 2020.
- [9] S. Mishra, H. Thakkar, P. K. Mallick, P. Tiwari, and A. Alamri, "A Sustainable IoHT based computationally intelligent healthcare monitoring system for lung cancer risk detection," *Sustainable Cities and Society*, vol. 72, Article ID 103079, 2021.
- [10] S. Chen, J. Wang, L. Zhang et al., "When internet of things meets e-health: an indoor temperature monitoring and control approach," *IEEE Internet of Things Magazine*, vol. 4, no. 3, pp. 12–16, 2021.
- [11] R. Johari, R. Tripathi, I. Kaur, K. Gupta, and R. K. Singh, "Roset," in *IoT Security Paradigms and Applications*, pp. 43–67, CRC Press, Boca Raton, FL, USA, 2020.
- [12] R. Arul, R. Alroobaea, U. Tariq, A. H. Almulih, F. S. Alharithi, and U. Shoaib, "IoT-enabled healthcare systems using block chain-dependent adaptable services," *Personal and Ubiquitous Computing*, vol. 2, pp. 1–15, 2021.

- [13] B. S. Yıldız, "Optimal design of automobile structures using moth-flame optimization algorithm and response surface methodology," *Materials Testing*, vol. 62, no. 4, pp. 371–377, 2020.
- [14] M. Tariq, T. Sato, G. Srivastava, V. Marojevic, and M. Goldenbaum, "IEEE access special section editorial: lightweight security and provenance for internet of health things," *IEEE Access*, vol. 9, pp. 67501–67503, 2021.
- [15] T. A. Nguyen, I. Fe, C. Brito et al., "Performability evaluation of load balancing and fail-over strategies for medical information systems with edge/fog computing using stochastic reward nets," *Sensors*, vol. 21, no. 18, p. 6253, 2021.
- [16] A. M. Borujeni, M. Fathy, and N. Mozayani, "Developing and evaluating a real-time and cost-efficient architecture for COVID-19 using internet of health things," 2021.
- [17] Z. Ahmadi, M. Haghi Kashani, M. Nikravan, and E. Mahdipour, "Fog-based healthcare systems: a systematic review," *Multimedia Tools and Applications*, vol. 3, pp. 1–40, 2021.
- [18] R. H. Saeed, H. I. Dino, L. M. Haji, D. M. Hamed, H. M. Shukur, and O. H. Jader, "Science and business," *International Journal*, vol. 5, no. 1, pp. 115–126, 2021.
- [19] L. Shu, V. Piuri, C. Zhu, X. Chen, and M. Mukherjee, "IEEE access special section editorial: convergence of sensor networks, cloud computing, and big data in industrial internet of things," *IEEE Access*, vol. 8, pp. 210035–210040, 2020.
- [20] S. S. Kumar and M. S. Koti, "An hybrid security framework using internet of things for healthcare system," *Network Modeling Analysis in Health Informatics and Bioinformatics*, vol. 10, no. 1, pp. 1–10, 2021.
- [21] A. Asghar, A. Abbas, H. A. Khattak, and S. U. Khan, "Fog based architecture and load balancing methodology for health monitoring systems," *IEEE Access*, vol. 9, pp. 96189–96200, 2021.
- [22] S. M. Nagarajan, V. Muthukumaran, R. Murugesan, R. B. Joseph, M. Meram, and A. Prathik, "Innovative feature selection and classification model for heart disease prediction," *Journal of Reliable Intelligent Environments*, vol. 1, pp. 1–11, 2021.
- [23] A. Saiyeda, "Challenges and opportunities with green and sustainable computing in healthcare," in *Green Automation for Sustainable Environment*, pp. 19–35, CRC Press, Boca Raton, FL, USA, 2020.
- [24] C. Gopalakrishnan and M. Iyapparaja, "Multilevel thresholding based follicle detection and classification of polycystic ovary syndrome from the ultrasound images using machine learning," *International Journal of System Assurance Engineering and Management*, vol. 5, pp. 1–8, 2021.
- [25] N. Malik and S. K. Malik, "Using IoT and semantic web technologies for healthcare and medical sector," *Ontology-Based Information Retrieval for Healthcare Systems*, Wiley-Scrivener, Austin, TX, USA, pp. 91–115, 2020.
- [26] R. Manikandan, R. Patan, A. H. Gandomi, P. Sivanesan, and H. Kalyanaraman, "Hash polynomial two factor decision tree using IoT for smart health care scheduling," *Expert Systems with Applications*, vol. 141, Article ID 112924, 2020.