International Journal of Computer and Engineering Optimization (IJCEO)
Volume 02, Issue 04, July – August (2024)

RESEARCH ARTICLE

DWARF MONGOOSE OPTIMIZATION BASED TASK ALLOCATION MODEL IN CLOUD COMPUTING FOR COST EFFECTIVENESS

S. Lokesh [1, *] and M. Ramya Devi [2]

[1] Associate Professor, Department of Computer Science and Engineering, PSG Institute of Technology and Applied Research, Coimbatore, Tamil Nadu, India

[2] Assistant Professor in the Department of Computer Science and Engineering at Hindusthan College of Engineering and Technology, Coimbatore, Tamil Nadu, India

[*] *Corresponding e-mail: lokesh@psgitech.ac.in*

Abstract – Task scheduling plays an important role in the cloud computing (CC) platform. The cloud allocates each task using different physical and virtual machines. One disadvantage of existing task scheduling algorithms in CC is their inefficiency in handling dynamic workloads, leading to resource underutilization or overload. This can degrade performance and increase operational costs. To overcome this challenge, a meta-heuristic model for task scheduling for cloud called Dwarf Mongoose Algorithm based Task Allocation (DMA-TA) has been introduced. The proposed model intends to solve the issues of optimal resource allocation (ORA) and scheduling in Cloud model, using a parallel scheduling process can improve the task scheduling during the connectivity between serial operations remains constant. The significant factor of dynamic tasks is provided by users in serial manner as queue, in which the tasks are having different priorities based on the order of execution. Moreover, the model developed a Deadline-based Task Classification (DTC) for efficient results on scheduling. The paper uses dwarf mongoose optimization (DMO) for scheduling tasks in minimal latency, through which the user satisfaction is improved. The results are effectively analyzed based on parameters such as makespan, time effectiveness, cost effectiveness and efficiency. The proposed DMA-TA technique achieves the lowest processing time of approximately 10 ms, compared to BSO-LB, QODA-LB and CSLBA respectively.

Keywords *– Task scheduling, cloud computing, Deadline, Dwarf mongoose optimization.*

1. INTRODUCTION

Cloud computing (CC) is an advanced computing paradigm for varied applications that are connected to private systems or public networks for providing a dynamically scalable model for applications, databases and storage [1]. CC can be defined as the model that provides users with dynamic virtualization of resources that can be provided to the required users in a distributed manner [2]. Task Scheduling and Task Allocation play a significant role in the performance of CC models [3]. The process of scheduling the tasks in a cloud model with minimal latency is an important issue considered in Cloud, also for ORA and cost-effectiveness [4]. A well-executed task scheduling strategy allocates incoming requests (tasks) in a way that maximizes the allocation of available resources. Users should place their service requests online since CC is a technology that provides services via Internet [5]. An important CC issue is effective task scheduling. The practical usage of resources may be achieved via the use of proper task scheduling [6].

Additionally, the load balancing (LB) process handles the Service Level Agreements (SLAs), which are agreements between cloud users and the CSP. Virtual machines (VMs) or physical hosts may be used for LB in cloud computing. All nodes (hosts or virtual machines) receive an equal share of the dynamic workload thanks to this balancing technique [7]. LB algorithms come in two varieties: static and dynamic. The majority of stable environments with homogeneous systems are suitable for static-based balancing techniques. In environments that are both homogenous and heterogeneous, dynamic-based balancing methods are more flexible and efficient. However, compared to dynamic LB techniques, the use of static LB procedures has lower system overhead [8].

A load is the distribution of different tasks among VMs in a CC environment. The LB problem can be defined in a number of ways. Assigning a finite number of tasks to different Physical Machines (PMs), which are in turn assigned to different VMs of each PM. LB algorithm effectiveness is determined by the efficiency of task allocation to the cloud [9]. The existing drawback that motivates the proposed to introduces the DMA-TA algorithm. The proposed model intends to solve the issues of ORA and scheduling in Cloud model, using a parallel scheduling process can improve the task scheduling during the connectivity between serial operations remains constant. The key contributions of this work are summarized as,

- A DMA-TA is proposed to solve the issues of ORA and scheduling in the Cloud model, using a parallel scheduling process can improve the task scheduling during the connectivity between serial operations remains constant.

- The significant factor of dynamic tasks is serially provided by users as a queue, in which the tasks are having different priorities based on the order of execution.
- The model developed a DTC for efficient results on schedule. The model uses DMO for scheduling tasks in minimal latency, through which user satisfaction is improved.
- The results are effectively analyzed based on parameters such as makespan, time effectiveness, cost effectiveness and efficiency.

The structure of the paper is organized as follows, section-2 describes the literature survey, the proposed DMA-TA was explained in section-3, the performance outcomes and their comparison analysis were provided in section-4 and section-5 encloses with conclusion and future work.

2. LITERATURE SURVEY

The related work carried out by various researchers on cloud infrastructure, scheduling in cloud environment and optimization of scheduling algorithms. A classification is made on various research work based on heuristic- based, meta-heuristic-based and hybrid model of framework. In this section, the research gap is identified in line with our motivation and objective of our study.

In 2019, Vijayakumar et al. [10] proposed a Dragonfly Optimization algorithm. This algorithm provides LB based on the constraint measures. The capacity of the VM and load are calculated and when the VM load is higher than the recommended value, a given job is reallocated with relatively lower load. Loads of the VM are balanced using capacity and applied loads in each virtual machine.

In 2020, Junaid, M.et al [11] propose a Data File Type Formats (DFTF) for LB. The approach is a modified form of Cat Swarm Optimization that uses the SVM algorithm. The SVM is a classifier that can differentiate between a variety of inputs, including text, images, and audio and video. Simulation findings showed improved response time (8.2%), SLA violation (8.9%), migration time (13%), throughput (7%), optimization time (9.7%), energy consumption (8.5%), overhead time (6.2%), and average execution time (9%), especially when compared to prior techniques.

In 2021, Mishra, K., and Majhi, S.K. [12] have introduced the approach known as Bird Swarm Optimization Load Balancing (BSO-LB). This algorithm views the VM as the food patches and the tasks as the birds. The algorithm is created by mimicking the flock of birds' actions. According to the experimental results, the recommended methods outperform alternative algorithms.

In 2022, Latchoumi, T.P., and Parthiban, L., [13] introduced the Quasi Oppositional Dragonfly Algorithm for LB (QODA-LB) for ORA. In the suggested QODA-LB technique, an OF is calculated based on three variables: load, run time, and running costs. Simulation results outperformed leading methods and demonstrated maximum LB effectiveness.

In 2022, Bal, P.K., et al. [14] presented a method for improving efficiency in CC by combining resource allocation security with resource allocation employing hybrid machine learning (RATS-HM). Compared to existing techniques, the suggested strategy leverages memory resources and reduces energy consumption more effectively.

In 2021, Singh et al. [15] presented a Crow Search Based Load Balancing Algorithm (CSLBA) to address the problematic of mapping from task to resource. This method is compared with standard Ant Colony Optimization. In comparison to other existing optimization methods, CSLBA has produced good performance and has identified as the optimum LB method.

In 2020, Shudong Wang et al. [16] presented an algorithm based on catastrophic genetic algorithm (CGA) that work for CC infrastructure to schedule the task for edge devices to achieve a globally optimum solution. The completion time of the task and the penalty factor are used as the feature of the algorithm. The advanced roulette selection technique, an optimized mutation and the crossover procedure have been applied to increase the effectiveness of the procedure.

In 2025, Mishra, P.K. and Chaturvedi, A.K., [17] suggested a Laxity-based Cost-efficient Task Scheduling (LCTS) to address modern task scheduling challenges, like maximizing energy use while balancing cost and delay. According to the results, the suggested approach outperformed Round Robin (RR) and GA in terms of execution time and cost. The simulation findings demonstrated a 6.99%-17.36% reduction in cost and a 4.58%-9.09% reduction in execution time when compared to GA and RR, respectively.

However, considering the load of VM resources during the task scheduling eliminates unnecessary overheads and cost in maximizing resource utilization. Moreover, it is found that, hybrid algorithms are more effective than other stand-alone algorithms in improving various performance parameters. The paper proposes hybrid algorithms for DMA-TA scheduling in CC environments with an account on other performance parameters such as response time, cost and energy efficiency.

3. PROPOSED MODEL

This section describes about the working process of DMA-TA model. The proposed model framework in allocating available VM in efficient manner is provided in Figure 1, using dwarf mongoose optimization.

3.1 Task scheduling

As the Cloud has large pool of resources, logical task scheduling takes place for easy resource access. Resource management and scheduling models are used for minimizing the latency and provide more user satisfaction. Based on the task types, the classification happens as, serial and parallel tasks. Dynamic scheduling policies are used for determining fairness of schedule process, through which the model efficiency is enhanced. For enhancing the results of task scheduling in cloud, deadline based intellectual optimization methodologies are applied to the input parallel tasks.

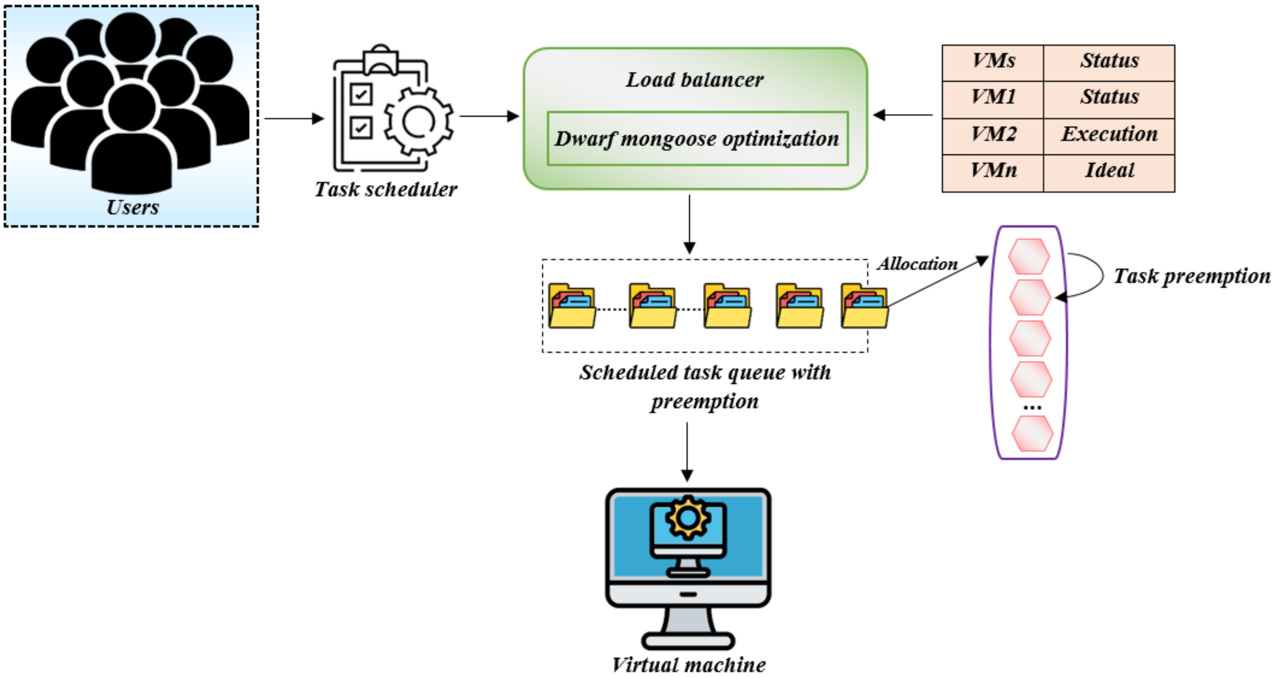

Figure 1. DMA-TA Framework

3.2 Deadline-based Task Classification (DTC)

In this model, the set of tasks are considered for performing each, which is from the system, given as, $D_N = \{1,2,3, \ldots., N\}$. The tasks for the system include image processing, location models, interactive gaming and so on. Those tasks are to be completed in deadline. Each task is considered to be having three features, which can be given as,

$$TK_i = [dd_i, S_i, exp\, T_i]$$

$$i\epsilon N \quad (1)$$

For task 'TK_i', 'S_i' is the input data size, comprises of input files and, 'exp T_i' deadline of the task to be completed. The length of the task is mentioned as, 'dd_i'. Further, task classification is carried out based on the features for executing that on to the cloud. Here, task sensitivity is defined on the basis of ratio between the task delay and the length of the task. The task scheduling is mainly for minimizing the task execution time.

Here, p_i^c denotes the computing power that is provided to each task by the device. Hence, the time for tasks local execution is computed as,

$$T_{local}^i = \frac{dd_i}{p_i^c} \quad (2)$$

Further, the time computed for Cloud is given as,

$$T_{trans}^i = \frac{S_i}{rate} \quad (3)$$

Rate is defined as the number of tasks that are assigned to the cloud. The rate of upload is considered as the fixed value. For performing consecutive task scheduling in cloud, the tasks are to be ordered based on the sensitivity. The sensitivity rate can be defined as,

$$Sen_{T_i} = \frac{dd_i}{exp\, T_i} \quad (4)$$

The process of task classification is given Table 1. The task scheduling process involves in ordering the input tasks to the multiple VM in optimal manner with cost and time effectiveness. For that, the following points are considered.

i. The tasks are not inter-linked with one another

ii. The task size and the computing efficiency of the VMs are well-known,

3.3 Dwarf mongoose optimization

The DMO algorithm is essentially a bio-inspired optimization technique that has been used for load balancing. It takes cues from the social relationships and behaviour of African dwarf and small mongooses. These little mammals are renowned for their cooperative behaviours and highly structured social structure, both of which are essential to their survival in the wild. Cooperative foraging, in which dwarf mongooses forage in groups while exchanging data about potential food sources, is a crucial component of their behaviour. In order to locate and take use of food supplies, the group effectively investigates their surroundings recognitions to their collective searching approach. Some mongoose members serve as sentinels, alerting the group to danger and keeping watch for predators in addition to foraging. Furthermore, dwarf mongooses' dynamic and adaptable division of labor, which includes grooming, sentinel duty, and foraging, enables the group to adapt well to shifting environmental conditions. The DMO is made to better explore and utilize the search space by combining these cooperative and sociable behaviors, striking a balance between the two.

Swarm intelligence-based optimization methods often start with the herd or tribe being initialized at random. Similarly, as in Equation (12), the DMO optimization starts with initializing the mongoose (M) candidate population.

$$M = \begin{bmatrix} m_{1,1} & \cdots & m_{1,y} \\ \vdots & \ddots & \vdots \\ m_{x,1} & \cdots & m_{x,y} \end{bmatrix} \tag{12}$$

Where M is the current population assembly of mongoose candidates, which is created at random using Equation (12), x is the size of the entire population, y is the problem dimension, and M marks the location of the z^{th} dimension of the p^{th} population. Equation (13) creates a stream of randomly dispersed integers that are uniformly distributed using built-in function unifrnd. Furthermore, D denotes the number of decision variables or the features of the mongooses, LB and UB stand for the problem's bottom and upper bounds, respectively.

$$m_{p,z} = unifrnd\ (LD, UB, D) \tag{13}$$

Every preparation's best-fit outcome is regarded as the most optimal outcome to date. The degree of fitness, as stated in Equation (14), is then calculated for each mongoose herd member following the completion of initiation. The member selected as the alpha female has the highest fit probability.

$$\propto = \frac{Fit_n}{\sum_{n=1}^{x} Fit_n} \tag{14}$$

The peep sound, represented by the vocalization of the dominant female P, is what keeps the herd on the intended course. The modified technique for the solution was calculated using Equation (15):

$$S_{n+1} = S_n + \emptyset * P \tag{15}$$

Here, the distributed random number is denoted by $\emptyset$. For each iteration, the sleeping mound ∂ was calculated using Equation (16):

$$\partial_n = \frac{Fit_{n+1} - Fit_n}{max\{|Fit_{n+1}, Fit_n|\}} \tag{16}$$

The average of ∂n was calculated using Equation.

$$\gamma = \sum_{n=1}^{x} \partial_n \tag{17}$$

When the babysitter criteria were met, the algorithm moved on to the next group.

Scout group

According to Equation (18) a new sleeping mound was revealed if the family silages far enough during this period.

$$S_{n+1} = \begin{cases} S_n - CVM * \emptyset * rand\left[S_n - \overrightarrow{M_v}\right] & if\ \theta_{n+1} < \theta_n \\ S_n + CVM * \emptyset * rand\left[S_n - \overrightarrow{M_v}\right] & else \end{cases} \tag{18}$$

The rand value in this case was created between [0, 1]. $\overrightarrow{M_v}$ was the movement vector, while CVM was a parameter used to regulate the mongoose group's collective violative movement. Both were computed using Equations (19) and (20):

$$CVM = \left(1 - \frac{i}{MAX_i}\right)^{\left(\frac{2i}{Max_i}\right)} \tag{19}$$

$$\overrightarrow{M_v} = \sum_{n=1}^{x} \frac{S_n \times \partial_n}{S_n} \tag{20}$$

The lower-ranking individuals who stay with the children are the babysitters. The remainder of the tribe engages in everyday hunting excursions and extra foraging activities while babysitters are rotated to help the alpha female. Based on the predicted schedule, the proposed algorithm allocates the model.

4. RESULTS AND DISCUSSIONS

This section describes data evaluation to a better understanding the model performance of required heuristic model with respect to the compared works based on significant scheduling parameters, such as, makespan, scheduling length, processing time and efficiency. For evaluating the proposed model under different cases, the communication to computational ratio factor is used to make different datasets with respect to the processing flows and the results and comparisons are provided below.

i. Makespan:

Makespan is the prominent factor, which is determined from the total execution time. The equation is given in the following equation.

$$Makespan = min\ \{max\ (EC(t_i))\} \tag{21}$$

The makespan based results are analyzed here based on the resources and the number of available VM. Table 2 compares the makespan with existing methods such as BSO-LB [12], QODA-LB [13] and CSLBA [15]. BSO-LB [12], QODA-LB [13] and CSLBA [15] based on varying number of VM. Here, the VM are varied from 20 to 40. From, the table, it is clear that the proposed method achieves lesser makespan when associated to other existing techniques.

Table 2. Makespan Analysis based on VMs

Models	20	25	30	35	40
BSO-LB [12]	3,570	4,124	5,057	5,105	5,593
QODA-LB [13]	4,214	4,630	4,431	5,599	4,828
CSLBA [15]	1,981	2,619	3,648	4,064	4,564
DMA-TA	1,108	1,385	1,114	783	1,216

ii. Scheduling length

This factor Scheduling length is a normalized factor for measuring model efficiency. When the scheduler executes the tasks significant to the decisive path on the fastest VM, the makespan factor cannot be lesser than the path length. The comparisons on time for execution is evaluated based scheduling length based on tasks and VM.

$$Scheduling_{length} = \frac{makespan}{\sum_{t_i \epsilon path} \min\ (ET(t_i, VM_k))} \tag{22}$$

By effective scheduling using DMO model, the time for processing is minimal when compared with other models.

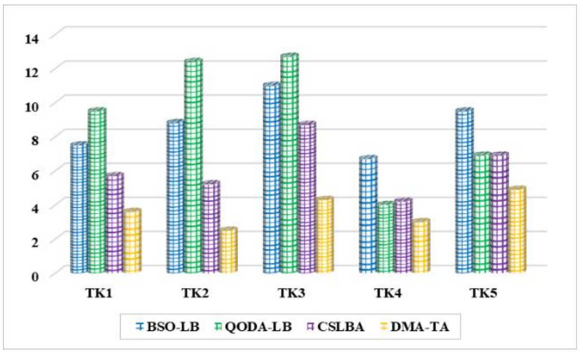

Figure 3. Results for scheduling time based on Tasks

Figure 3 illustrates the comparison of proposed DMA-TA method with existing methods in terms of scheduling time based on tasks.

iii. Processing Time:

The evaluation factor processing time is a significant conceit in parallel programming for attaining minimal time in processing. Figure 4 shows the comparison of processing time. It is measured with the following equation.

$$processing\ time = \frac{Execution\ of\ tasks\ with\ fastest\ VM}{makespan} \quad (23)$$

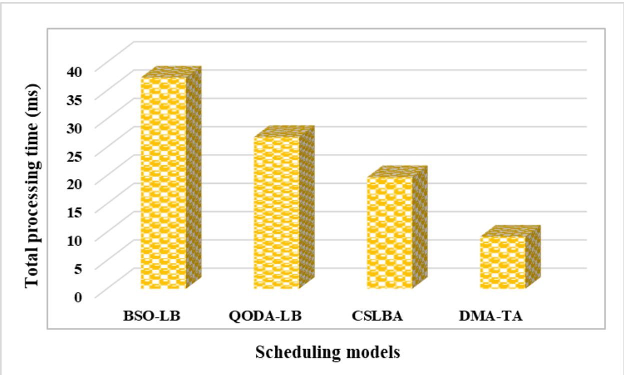

Figure 4. Processing Time Comparisons

iv. Model Efficiency:

The model efficiency is defined mainly based on the cost effectiveness and processing time. This factor is evaluated as,

$$Model\ Efficiency = \frac{processing\_time}{No.\ of\ used\ VMs} * 100\% \quad (24)$$

Table 3. Results for Cost Effectiveness

Models	TK1	TK2	TK3	TK4	TK5
BSO-LB [12]	75.41	68.46	55.23	73.35	82.24
QODA-LB [13]	54.62	60.18	65.68	65.08	54.14
CSLBA [15]	61.81	49.18	49.73	54.14	59.03
DMA-TA	32.69	31.06	29.43	44.83	35.53

Further, cost effectiveness is calculated and the results are provided in Table 3. The outcomes are shown that the proposed model achieves minimal cost by effectively utilizing the cloud resources.

5. CONCLUSION

This research proposed a meta-heuristic model for task scheduling in cloud called DMA-TA. The model proposes a Deadline based Task classification and task scheduling using DMO. The main objective of the model is to reduce the processing time and computational cost with maximal efficiency in processing input tasks with available VMs. The results are effectively analyzed based on parameters such as makespan, time effectiveness, cost effectiveness and efficiency. The results evidenced the model efficiency openly, providing better results than the compared works. The proposed DMA-TA technique achieves the lowest processing time of approximately 10 ms, compare to BSO-LB, QODA-LB and CSLBA respectively. In future, the LB can be done using some other techniques such as swam intelligence technique, and the work can also be extended in scientific workflows.

CONFLICTS OF INTEREST

The authors declare that there is no conflict of interest.

FUNDING STATEMENT

Authors did not receive any funding.

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] Z.A. Khan, I.A. Aziz, N.A.B. Osman and S. Nabi, "Parallel Enhanced Whale Optimization Algorithm for Independent Tasks Scheduling on Cloud Computing", *IEEE Access*, vol. 12, pp. 23529-23548, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[2] S. Sharma, and N. Kumar Pandey, "Multi-faceted job scheduling optimization using Q-learning with ABC in cloud environment", *International Journal of Computing and Digital Systems*, vol. 15, no. 1, pp. 1-16, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[3] A. Satouf, A. Hamidoğlu, Ö.M. Gül, A. Kuusik, L. Durak Ata, and S. Kadry, "Metaheuristic-based task scheduling for latency-sensitive IoT applications in edge computing", *Cluster Computing*, vol. 28, no. 2, pp. 1-17, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[4] M. Zambianco, S. Cretti, and D. Siracusa, "Disruption-aware Microservice Re-orchestration for Cost-efficient Multi-cloud Deployments", 2025. [CrossRef] [Google Scholar] [Publisher Link]

[5] A. Jenice Prabhu, S. Vallisree, S. N. Kumar, R. Sitharthan, M. Rajesh, A. Ahilan, M. Usha, "Efficient Data Security Using Hybrid RSA-TWOFISH Encryption Technique on Cloud Computing", *In International Conference on Frontiers of Intelligent Computing: Theory and Applications Singapore: Springer Nature Singapore*. pp. 495-502, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[6] O.L. Abraham, M.A. Ngadi, J.B.M. Sharif, and M.K.M. Sidik, "Multi-Objective Optimization Techniques in Cloud Task Scheduling: A Systematic Literature Review". *IEEE Access*, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[7] M. Jesi, A. Appathurai, M. Kumaran, A. Kumar, "Load Balancing in Cloud Computing via Mayfly Optimization Algorithm". *Revue Roumaine Des Sciences Techniques—Série*

Électrotechnique Et Énergétique. vol. 69, no. 1, pp. 79-84, 2024. [CrossRef] [Google Scholar] [Publisher Link]
[8] M. Hamdan, E. Hassan, A. Abdelaziz, A. Elhigazi, B. Mohammed, S. Khan, A.V. Vasilakos, and M.N. Marsono, "A comprehensive survey of load balancing techniques in software-defined network", *Journal of Network and Computer Applications*, vol. 174, pp. 102856, 2021. [CrossRef] [Google Scholar] [Publisher Link]
[9] S.K. Mishra, B. Sahoo, and P.P. Parida, "Load balancing in cloud computing: a big picture", *Journal of King Saud University-Computer and Information Sciences*, vol. 32, no. 2, pp.149-158, 2020. [CrossRef] [Google Scholar] [Publisher Link]
[10] V. Vijayakumar, and K.S. Joseph, "Adaptive load balancing schema for efficient data dissemination in Vehicular Ad-Hoc Network VANET", *Alexandria Engineering Journal*, vol. 58, no. 4, pp. 1157-1166, 2019. [CrossRef] [Google Scholar] [Publisher Link]
[11] M. Junaid, A. Sohail, R.N.B. Rais, A. Ahmed, O. Khalid, I.A. Khan, S.S. Hussain, and N. Ejaz, "Modeling an optimized approach for load balancing in cloud", *IEEE Access*, vol. 8, pp. 173208-173226, 2020. [CrossRef] [Google Scholar] [Publisher Link]
[12] K. Mishra, and S.K. Majhi, "A binary bird swarm optimization-based load balancing algorithm for cloud computing environment". *Open Computer Science*, vol. 11, no. 1, pp. 146-160, 2021. [CrossRef] [Google Scholar] [Publisher Link]
[13] T.P. Latchoumi, and L. Parthiban, "Quasi oppositional dragonfly algorithm for load balancing in cloud computing environment". *Wireless Personal Communications*, vol. 122, no. 3, pp. 2639-2656, 2022. [CrossRef] [Google Scholar] [Publisher Link]
[14] P.K. Bal, S.K. Mohapatra, T.K. Das, K. Srinivasan, and Y.C. Hu, "A Joint Resource Allocation, Security with Efficient Task Scheduling in Cloud Computing Using Hybrid Machine Learning Techniques", *Sensors*, vol. 22, no. 3, pp. 1242, 2022. [CrossRef] [Google Scholar] [Publisher Link]
[15] H. Singh, S. Tyagi, and P. Kumar, "Cloud resource mapping through crow search inspired metaheuristic load balancing technique". *Computers & Electrical Engineering*, vol. 93, pp. 107221, 2021. [CrossRef] [Google Scholar] [Publisher Link]
[16] S. Wang, Y. Li, S. Pang, Q. Lu, S. Wang, and J. Zhao, "A task scheduling strategy in edge-cloud collaborative scenario based on deadline", *Scientific Programming*, 2020. [CrossRef] [Google Scholar] [Publisher Link]
[17] P.K. Mishra, and A.K. Chaturvedi, "An Improved Laxity based Cost Efficient Task Scheduling Approach for Cloud-Fog Environment', *Engineering, Technology & Applied Science Research*, vol. 15, no. 1, pp. 19037-19044, 2025. [CrossRef] [Google Scholar] [Publisher Link]

AUTHORS

Lokesh S got the B.E., in Computer Science and Engineering in 2005 from Anna University, M.E., Degree in Computer Science and Engineering from Anna University in 2007 and Ph.D., in Information and Communication Engineering in 2015 from Anna University, respectively. He is having 15 years of teaching experience and currently associate as Associate Professor, Department of Computer Science and Engineering, PSG Institute of Technology and Applied Research, Coimbatore. His research areas are Human Computer Interaction, Speech Recognition, Data Analytics and Machine Learning.

Ramya Devi M is Assistant Professor in the Department of Computer Science and Engineering at Hindusthan College of Engineering and Technology, Coimbatore. She got Diploma in Computer Networks from PSG Polytechnic College in 2003, B.Tech. Degree in Information Technology from Anna University in 2008 and M.E. Degree in Computer Science and Engineering in 2011 from Karpagam University, respectively. Her research area is Vehicular Cloud Computing, Human Computer Interaction and Wireless Networks.

Arrived: 09.07.2024
Accepted: 11.08.2024