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

RESEARCH ARTICLE

DEADLINE AWARE TASK SCHEDULING USING DRAGONFLY OPTIMIZATION IN CLOUD ENVIRONMENT

R.R. Sathiya [1, *] and A. Ahilan [2]

[1] Department of Computer Science and Engineering, Amrita School of Computing, Amrita Vishwa Vidyapeetham, Coimbatore, India

[2] Department of Electronics and Communication Engineering, PSN College of Engineering and Technology, Tirunelveli, Tamil Nadu, India

[*] *Corresponding e-mail: rr_sathiya@cb.amrita.edu*

Abstract – Cloud computing has emerged as the more alluring option, when compared to grid computing. Cost financial criteria must be taken into account while scheduling the jobs effectively, and performance under user-defined limits must be optimised during execution time. To create a load-balancing strategy in cloud settings, to determine the problems that impact LBC. To overcome this challenge Dragonfly Optimization algorithm-based Task allocation in Cloud (DOT-Cloud) is proposed that can schedule jobs in a heterogeneous cloud environment with competing time and cost Quality of Service (QoS) criteria. The main goals of the proposed DOT-Cloud model are to decrease the makespan within the user-defined deadline and to reduce financial costs without going over the user-defined budget. Comparing the proposed DOT-Cloud model to state-of-the-art algorithms, simulated studies show that the proposed model performs better in terms of minimising the makespan and cost. The improvement of makespan ratio for our proposed DOT-Cloud over is QMPSO, DFTF and HLB algorithms is by 39%, 5%, and 41%, , in addition to the improvement of cost ratio for our proposed DOT-Cloud model over QMPSO, DFTF and HLB algorithms by 36%, 14%, and 38% respectively.

Keywords *– Cloud computing, load balancing, Dragonfly optimization algorithm, Quality of Service, and Makespan.*

1. INTRODUCTION

Cloud computing, which relies on distant computers that can be accessed online and decentralizes resources, radically changes conventional ideas of data processing and storage. Because sensitive data travels across networks and is housed on infrastructures run by outside providers, this change creates a complicated web of security flaws [1]. Concerns regarding data availability, confidentiality, and integrity in a setting vulnerable to various attacks temper the appeal of cloud services [2]. The cloud computing paradigm was developed using the following four deployment models: public, private, community, and hybrid cloud. In a private model, infrastructure is only accessible to a single company with several users, some of whom may be third parties [3]. A community model is offered for a certain group of that organization's clients. The broader public has access to the public model. It is situated on the cloud provider's workspace. Combining two or more distinct cloud deployment strategies is known as a hybrid deployment model [4]. Data and application portability is made possible by standardized or patented technologies, which push these together.

Workload balancing is a crucial component of cloud computing architecture that aids in the distribution of computational resources. The memory, storage, and computing speed of each virtual machine vary. The sole method for mapping a task to an ideal virtual machine (VM) that prevents any VM from being overwhelmed is load balancing [5]. A load balancing divides the workload across several users across one or more workstations, servers, networks, or other IT resources. The traditional load balancing design is entirely different from this cloud method. Several academics are working on various sorts of optimal resource approaches in the cloud space all around the world. Utilizing run time dispersion, the corresponding method enhances performance and better balances [6]. For heterogeneous cloud computing to provide optimal resource utilization and quality of service, load balancing is necessary. For the best possible resource usage and customer satisfaction at the lowest possible cost, load balancers help ensure that resources are distributed fairly and equally among workloads [7].

There are several problems with the current load-balancing techniques that require quick fixes. In addition to load balancing, there is an issue with resource management, energy conservation, carbon emissions, QoS, VM migration, execution time, and VM performance [8]. During task computing, one of the primary issues that prevents virtual machines from experiencing overloading or underloading is load balancing in the cloud (LBC) [9,10]. To create a load-balancing strategy that works in cloud settings, it is necessary to determine the problems. To solve the problem, we proposed a DOT-Cloud technique. The major contributions are given below:

- This paper proposed a DOT-Cloud to load balance the work in cloud and increase the importance of service security, deadline and task scheduling.
- Developing task scheduling algorithms that prioritize deadlines, budget constraints, security, and system dependability to meet Service Level Agreement requirements.
- Balancing workload distribution to maximize resource usage and prevent underutilization using proposed DOA.
- The proposed DOT-Cloud decrease the makespan within the user-defined deadline and to decrease financial costs without going over the user-defined budget.

The structure of the paper is organised as follows, section-2 defines the literature survey, the proposed method was explained in section-3, the performance results and their comparison analysis were provided in section-4 and section-5 encloses with result and discussion

2. LITERATURE SURVEY

In recent years, several researches have investigated the load balancing in cloud computing. This section that provides a review of some current research papers.

In 2020, Junaid, M.et al [11] proposed a Data File Type Formats (DFTF) for load balancing. The algorithm has employed the SVM algorithm and is a changed version of Cat Swarm Optimization. The SVM is a classifier distinguishing between the diverse inputs such as video audio, text and pictures. The updated Cat Swarm Optimization method distributes the job across the virtual machines based on the input. When compared to previous methodologies, simulation results revealed 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 %).

In 2020, Xiashuang et al. [12] proposed a Hybrid Load Balancing (HLB) scheduling algorithm by combining static method and dynamic priority. A scheduling technique is critical for efficiently performing jobs in a simulation model. A novel simulation model couldn’t be employed with the conventional blade-based parallel engine system. The authors additionally prioritize the concept and system phases according to their operational cycle. The method balances the VMs while reducing task request waiting times, which increases VM throughput and boosts the performance of the real computer. They have demonstrated that their suggested technique outperforms other existing strategies by comparing it to the identified and existing algorithm.

In 2021, Muteeh, A., et al., [13] introduced a multi-resource load balancing algorithm (MrLBA) for cloud computing. The suggested method tries to shorten the process and save time while keeping the system in balance. Benchmark workflows are used to validate the technique. To minimise cost and execution time while maximising resource utilisation, a balancing load is maintained among resources using MrLBA.

In 2023, Al Reshan, M.S. et al., [14] proposed Swarm Intelligence (SI) as a cloud computing load balancing solution. The proposed technique combined GWO-PSO strategy that leverages the advantages of global optimization and quick convergence. These two methods improve system effectiveness and resource distribution, collaborating to address the load-balancing problem. When compared to other methods, the proposed technique's overall speed of response is 12% faster. Additionally, PSO is improved to 97.253% in terms of convergence by the most optimum value derived from the objective function of the proposed GWO-PSO method.

In 2025, Kumar, N.V., et al., [15] proposed a hybrid deep learning (DL)-based load balancing method. The deep embedding cluster (DEC) also takes into account the CPU, bandwidth, memory, processing components, and frequency scaling factors in determining a virtual machine (VM). The tasks finished on the overloaded virtual machine (VM) are allocated to the underloaded VM according to their value in order to provide cloud load balancing. The effectiveness of this strategy is further assessed using load, capacity, resource consumption, and success rate; optimal values of 0.147, 0.726, 0.527, and 0.895 are reached.

In 2021, Alagarsamy, M., et al. [16] proposed a cost-aware ant colony optimization-based (CACO) load balancing approach for reducing reaction time, execution time, and cost. The experimental results shows that the suggested method decreases processing time, power consumption, response time, and cost. The suggested CACO framework decreases carbon footprint by 45% when compared to existing approaches.

3. PROPOSED METHODOLOGY

In this section, a novel DOT-Cloud has been proposed to schedule jobs in a heterogeneous cloud environment with competing time and cost QoS criteria while still ensuring user satisfaction. The main goals of the proposed DOT-Cloud model are to decrease the makespan within the user-defined deadline and to reduce financial costs without going over the user-defined budget. The scheduling DOT-Cloud model, as depicted in Figure 1, comprises cloud users that send their tasks together with the limitations (deadline, budget). Budget and deadline constraints will cause the work to be implemented in cluster one, deadline constraints will cause it to be implemented in cluster 2, and budget constraints will cause it to be implemented in cluster 3.

3.1. Deadline and Budget Constraints

The deadline and budget values must be negotiated between users and cloud service providers for meaningful scheduling solutions so that the agreed values are reasonable and mutually acceptable. Under the constraints of the deadline and budget, some metrics can work and fall within them such as makespan, number of violations, improvement of makespan ratio, improvement of cost ratio, total gain cost, provider profit, and resource utilization.

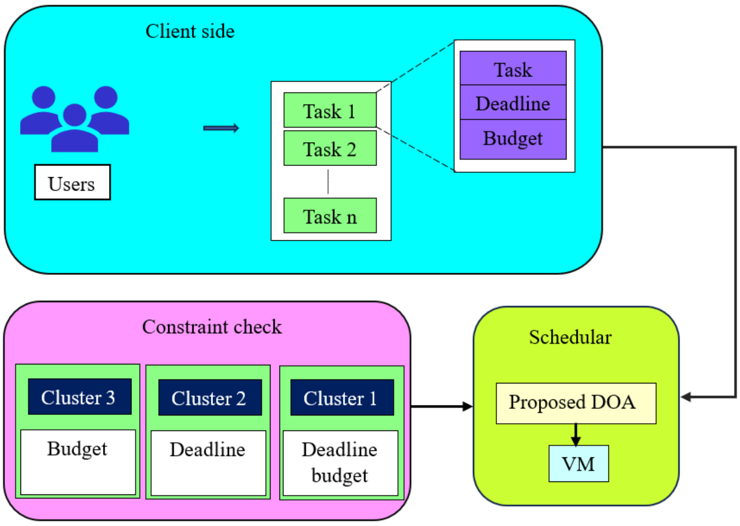

Figure 1. Overall block diagram for the proposed DOT-Cloud

3.2. Task Scheduling Based on Deadline Budget Model

This contribution concentrates on deadline and financial constraints in task scheduling to enhance the quality of service in association with two fundamental metrics, namely, minimize execution time and cost of executed tasks, and ultimately increase the revenue of services as well as resource utilization of the virtual machines and the host. Because of this, we proposed the DOT-Cloud to execute user tasks on virtual machines (VMs) while adhering to QoS restrictions more quickly. The DOT-Cloud adheres to a user-defined timeline and budget while attempting to reduce execution time and expense. Based on customer satisfaction, each task has a certain type of constraint. Three different types of constraints are assigned in our proposed DOT-Cloud model as shown in Equation 1:

$$Task\ constraint\ type = \begin{cases} d\ and\ b & if\ the\ constraints\ are\ deadline\ and\ budget \\ d & if\ the\ constraint\ is\ deadline \\ b & if\ the\ constraint\ is\ budget \end{cases} \quad (1)$$

3.3. Clustering of Resources

According to the user's satisfaction, the resources are grouped in the following ways:

1. The first cluster consists of a group of virtual machines that adhere to the financial and time limits.
2. The second cluster is made up of VMs that can complete tasks by the deadline.
3. The third cluster is made up of VMs that simply adhere to financial restrictions.

3.4. Dragon fly optimization

In this paper a proposed DOT-Cloud enhanced Dragonfly optimization to improves the convergence efficiency and solution accuracy. This algorithm inspired by the swarming behavior of dragonflies, helps detect the most related features, falling dimensionality, and improving model efficiency. An intelligent search-optimization method, the DOA is a type of evolutionary algorithm. The concept was inspired by dragonflies' both static and dynamic activity. An example of a static swarm is a tiny group of dragonflies that pursue prey in a small area. Fly movement is typified by sudden and quick changes in their individual flight trajectories. A dynamic swarm, on the other hand, is a sizable group of flies that move in a consistent direction over a considerable distance in order to migrate from one location to another.

The exploitation phase is characterized by the static behavior of swarm dragonflies (DFs), whereas the exploration phase is characterized by the dynamic behavior of swarm DFs, because the goal of these dragonflies in a swarm is analogous to the optimization issue. This provides the groundwork for DOA. For representing the movement of the flies in a cluster mathematically, five characteristics of dragonflies are discussed: separation, alignment, cohesiveness, food and enemy. Separation, alignment, cohesiveness, food, and adversary characteristics of an individual dragonfly in a cluster are represented by the Sep_i, Alg_i, Coh_i, A f_i, and Ee_i. Within the specified locality, the distance between neighbouring DFs is essential to optimize the search space and prevent collisions. Let i stand for the number of people in a cluster that has n neighbours. Equation (7) is used to display the Separation Sepi of the i individual in a cluster, as illustrated below. where x_k is the location of the k_{th} neighbouring DF and x is the present position of DF.

$$Sep_i = \sum_{k=1}^{n}(x - x_k) \quad (2)$$

Equation (8) uses the alignment term Alg_i to match the individual's velocities with those of other DFs in the same locality. The velocity of the kth neighbouring DF is denoted by V_k.

$$Alg_i = \frac{\sum_{k=1}^{n} V_k}{n} \quad (3)$$

Every DF individual in a cluster tends to travel toward the mass center of nearby DFs. Equation (9) determines the cohesiveness feature Coh_i of DF:

$$Coh_i = \frac{\sum_{k=1}^{n}(V_k)}{n} \quad (4)$$

Since food is necessary for survival, all of the people in a cluster tend to travel in its direction. The attractiveness of food Equation (5) is used to obtain a f_i feature at point x

$$Af_i = x_{food} - x \quad (5)$$

A cluster of people tends to disperse from the adversary. Equation (6) can be used to determine the enemy feature Ee_i at the enemy x_e position.

$$Ee_i = x_e + x \quad (6)$$

Combining these five characteristics affects how DFs behave within a cluster. Equation (7) represents the updated location of each distinct DF, which is determined by step Δxi.

$$x_i = x_i + \Delta x_i \quad (7)$$
$$\Delta x_i = w\Delta x_i + (a.Sep_i + b.Alg_i + c.Coh_i + d.Af_i + e.Ee_i \quad (8)$$

where d stands for the food factor and e for the enemy component. The inertial weight is w, and the alignment, separation, and cohesion weights are a, b, and c. A dragonfly's alignment, separation, food, cohesiveness, and adversary characteristics are represented by its Sep_i, Alg_i, Coh_i, Af_i and Ee_i. Various parameter values can be used to achieve the variety in the DFs' exploitative and exploratory behaviours. It improves the convergence efficiency and solution accuracy.

4. RESULTS AND DISCUSSIONS

We employed the CloudSim toolkit simulator for implementing the experiments of the DBS model. The CloudSim is characterized by modelling behaviour for components of cloud system like data center, processing elements, virtual machine, etc.

The CloudSim toolkit is a commonly used simulator due to its flexibility and simplicity, and is executed using java language. The advantages of the CloudSim are that it allows the establishment of policies to bind tasks with VMs, allocating VMs for hosting them in data centers, scheduling of VMs, and for energy consumption.

4..1. Performance Evaluation

In this section, the performance indicators that were employed to assess the suggested DOT-Cloud model has been discussed. Following are some measures that are employed in task scheduling to gauge how effective the algorithms are:

4.2. Makespan

The amount of time needed to perform the tasks must be decreased from the user's perspective. According to Equation 9, the makespan is the sum of the execution times for all jobs carried out by a given VM.

$$M = \sum_{x=1}^{m}(ECT_x * B[x, y]) \quad (9)$$

Where m is the total number of tasks, and 1 ≤ j ≤ m, B[x, y], a Boolean variable and M is the makespan.

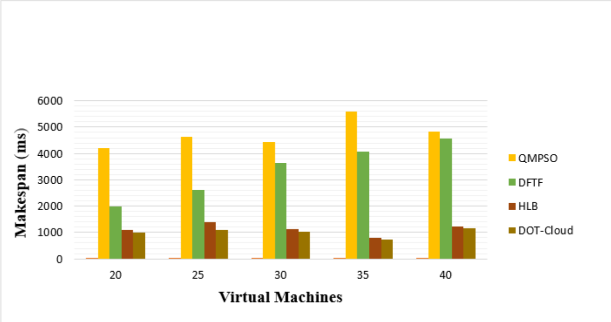

Figure 2. Comparison of makespan

Figure 2 represents the comparison of proposed DOT-Cloud technique with the existing techniques such as QMPSO, DFTF and HLB in terms of makespan. Here, the virtual machines are varied from 20 to 40. From, the figure, it is clear that the proposed method achieves lesser makespan when compared to other existing techniques.

4.3. Total Gain Cost

The capacity of an algorithm to carry out tasks within a certain budget limitation is measured by comparing the task's cost to its budget. This is another performance metric that is used to assess the effectiveness of our suggested DOT-Cloud model. Equation 10 is used to compute the total gain cost.

$$Total\ Gain\ Cost = \sum_{x=1}^{m} Gain\ cost_x \quad (10)$$

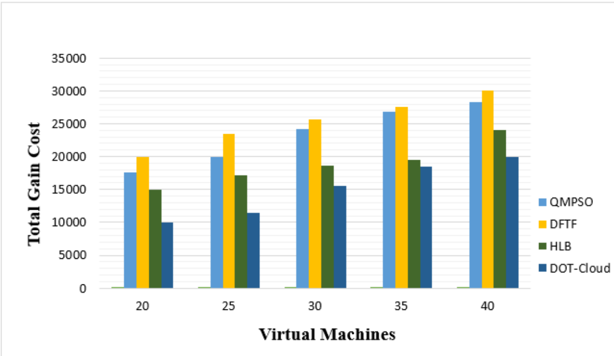

Figure 3. Comparison of total gain cost

Figure 3 represents the comparison of suggested DOT-Cloud technique with the existing techniques such as QMPSO, DFTF and HLB in terms of total gain cost. Here, the virtual machines are varied from 20 to 40. From, the figure, it is clear that the suggested technique achieves lesser gain cost in comparison with other existing techniques.

4.4. Provider Profit

Profit, which is computed by deducting the task's actual implemented cost from its budgeted cost, is the key performance indicator for service providers. Profit is the sum of all revenues from the amounts charged to users for successfully completed tasks. Based on Equation 11, the profit will be determined.

$$Provider\ Profit = (\sum_{x=1}^{m}(TB_x - Gain\ count_x)) \quad (11)$$

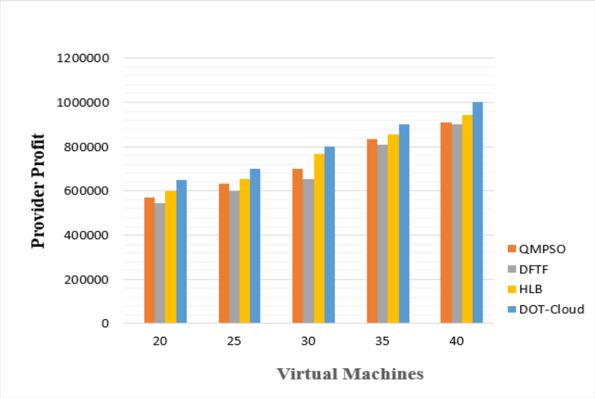

Figure 4. Comparison of Provider profit

Figure 4 represents the comparison of proposed DOT-Cloud technique with the existing techniques such as QMPSO, DFTF and HLB in terms of provider profit. Here, the virtual machines are varied from 20 to 40. From, the figure, it is clear that the suggested method achieves larger provider profit in comparison with other existing techniques.

4.5. Average Resource Utilization

Resource utilization is a crucial metric that the service provider is concerned about. The suggested DOT-Cloud model was evaluated with existing algorithms by the efficient exploitation of the resources by meeting the users' restrictions, where the resource utilisation should be maximised. Equation 12 gives a definition for the typical resource consumption.

$$Resource\ utilization = \frac{\sum_{b=1}^{m} number\ of\ successful\ tasks}{\sum_{t=1}^{n} S_d} \quad (12)$$

Table 1. Comparison of Resource utilization

Models	20	25	30	35	40
QMPSO	500	600	700	800	900
DFTF	450	540	650	750	850
HLB	540	650	760	856	950
DOT-Cloud	600	700	800	900	1000

Table 1 represents the comparison of proposed DOT-Cloud technique with the existing techniques such as QMPSO, DFTF and HLB in terms of resource utilization. Here, the virtual machines are varied from 20 to 40. It is clear from the table that the proposed method successfully utilises more resources than other methods currently in use.

4.6. Improvement of Cost Ratio

The cost ratio is a crucial statistic for evaluating how well the DOT-Cloud model is performing. Therefore, we calculated the improvement of cost ratio by the Equation 13.

$$cost\ ratio = \left(1 - \frac{\sum_{x=1}^{f} cost\ EOFA}{\sum_{x=1}^{f} cost\ of\ other\ algorithms}\right) \times 100 \quad (13)$$

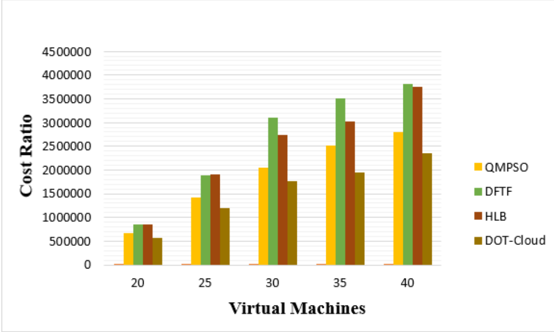

Figure 5. Comparison of cost ratio

Figure 2. represents the comparison of proposed DOT-Cloud technique with the existing frameworks such as QMPSO, DFTF and HLB in terms of total cost ratio. Here, the virtual machines are varied from 20 to 40. From, the table it is clear that the suggested techniques achieve lesser gain cost when compared to other existing techniques.

5. CONCLUSION

This paper proposed a task scheduling model to execute tasks while achieving user-defined constraints (deadline and budget). We propose a model called the deadline aware task scheduling using dragonfly optimization in cloud environment (DOT-Cloud) that can schedule jobs in a heterogeneous cloud environment with competing time and cost QoS criteria while still ensuring user satisfaction. The main goals of the suggested framework are to decrease the makespan within the user-defined deadline and to decrease financial costs without going over the user-defined budget. Comparing the suggested DOT-Cloud model to existing techniques, simulated studies show that the suggested model performs better in terms of minimising the makespan and cost. The proposed DOT-Cloud model is compared to the existing techniques: QMPSO, DFTF and HLB. The improvement of makespan ratio for our proposed DOT-Cloud model over QMPSO, DFTF and HLB algorithms is by 39%, 5%, and 41%, respectively, in addition to the improvement of cost ratio for our proposed DOT-Cloud model over QMPSO, DFTF and HLB algorithms by 36%, 14%, and 38%, respectively. Using a fault-tolerant scheduling system, we want to improve the functionality of DOT-Cloud in the future, allowing it to deal with resource breakdowns.

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] A.K.Y. Yanamala, "Emerging challenges in cloud computing security: A comprehensive review", *International Journal of Advanced Engineering Technologies and Innovations, ptible to diverse threats*, vol. 1, no. 4, pp. 448-479. [CrossRef] [Google Scholar] [Publisher Link]

[2] J. Zhu, Q. Li, S. Ying, and Z. Zheng, "Research on Parallel Task Scheduling Algorithm of SaaS Platform Based on Dynamic Adaptive Particle Swarm Optimization in Cloud Service Environment", *International Journal of Computational Intelligence Systems*, vol. 17, no. 1, pp. 260, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[3] N. Akhtar, B. Kerim, Y. Perwej, A. Tiwari, and S. Praveen, "A comprehensive overview of privacy and data security for cloud storage", *International Journal of Scientific Research in Science Engineering and Technology*, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[4] A. Choudhary, and R. Rajak, "A novel strategy for deterministic workflow scheduling with load balancing using modified min-min heuristic in cloud computing environment". *Cluster Computing*, vol. 27, no. 5, pp. 6985-7006, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[5] L. Abualigah, A.M. Hussein, M.H. Almomani, R.A. Zitar, H. Migdady, A.I. Alzahrani, and A. Alwadain, Improved synergistic swarm optimization algorithm to optimize task scheduling problems in cloud computing. Sustainable Computing: Informatics and Systems, 43, pp. 101012, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[6] R. Zhanuzak, M.A. Ala'Anzy, M. Othman, and A. Algarni, Optimising Cloud Computing Performance with an Enhanced Dynamic Load Balancing Algorithm for Superior Task Allocation. IEEE Access, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[7] P. Gupta, A. Anand, P. Agarwal, and G. McArdle, "Neural network inspired efficient scalable task scheduling for cloud infrastructure", *Internet of Things and Cyber-Physical Systems*, vol. 4, pp. 268-279, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[8] M. Jawhar, and H. Osman, "Quality of Service and Load Balancing in Cloud Computing: A Review". *AL-Rafidain Journal of Computer Sciences and Mathematics*, vol. 16, no. 1, pp. 15-22, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[9] D.A. Shafiq, N.Z. Jhanjhi, and A. Abdullah, "Load balancing techniques in cloud computing environment: A review". *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 7, pp. 3910-3933, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[10] U.K. Jena, P.K. Das, and M.R. Kabat, "Hybridization of meta-heuristic algorithm for load balancing in cloud computing environment", *Journal of King Saud University-Computer and Information Sciences*, 2020. [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] X. Wang, N. Li, G. Gong, X. Song, and Y. Guo, "Load-balancing scheduling of simulation tasks based on a static-dynamic hybrid algorithm", *Journal of Simulation*, vol.16, no. 2, pp. 182-193, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[13] A. Muteeh, M. Sardaraz, and M. Tahir, "MrLBA: multi-resource load balancing algorithm for cloud computing using ant colony optimization", *Cluster Computing*, vol. 24, no. 4, pp. 3135-3145, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[14] M.S. Al Reshan, D. Syed, N. Islam, A. Shaikh, M. Hamdi, M.A. Elmagzoub, G. Muhammad, and K.H. Talpur, "A fast converging and globally optimized approach for load balancing in cloud computing", *IEEE Access*, vol. 11, pp. 11390-11404, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[15] N.V. Kumar, S. Mohanty, and P.K. Pattnaik, "Hybrid algorithm for optimized clustering and load balancing using deep Q reccurent neural networks in cloud computing", *Bulletin of Electrical Engineering and Informatics*, vol. 14, no. 2, pp.1570-1578, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[16] M. Alagarsamy, A. Sundarji, A. Arunachalapandi, and K. Kalyanasundaram, "Cost-awareant colony optimization-based model for load balancing in cloud computing", *Int. Arab J. Inf. Technol.*, vol. 18. no. 5, pp. 719-729, 2021. [CrossRef] [Google Scholar] [Publisher Link]

AUTHORS

Sathiya RR completed her Bachelor's Degree in Information Technology from Bharathiar University, Coimbatore, Tamilnadu, India, in the year 2004 and her Master's Degree in Computer Science and Engineering from Anna University, Coimbatore, Tamilnadu, India, in the year 2010. Her areas of interest include cloud computing, Internet of Things, and Image processing. She has published technical papers in International Conferences and Journals.

A. Ahilan received Ph.D. from Anna University, India, and working as an Associate Professor in the Department of Electronics and Communication Engineering at PSN College of Engineering and Technology, India. His area of interest includes FPGA prototyping, Computer vision, the Internet of Things, Cloud Computing in Medical, biometrics, and automation applications. Served Guest editor in several journals of Elsevier, Benthom, IGI publishers. Also, have contributed original research articles in IEEE Transactions, SCI, SCIE, and Scopus indexed peer-review journals. He presented various international conference events like ASQED (Malaysia), ESREF(France). He is doing as a reviewer in IEEE Industrial Informatics, IEEE Access, Measurement, Multimedia Tools & Applications, Computer Networks, Medical systems, Computer & Electrical Engineering, neural computing and applications, Cluster Computing, IET Image Processing, and so on. He has IEEE and ISTE membership. He has worked as a Research Consultant at TCS, Bangalore, where he has guided many computer vision projects and Bluetooth Low Energy projects. Meanwhile, special Guest lectures, Practical workshops, Hands-on programming in MATLAB, Verilog, and python at various technical institutions around India.

Arrived: 02.07.2024
Accepted: 05.08.2024