IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 15, No. 4, August 2026, pp. 3683~3691
ISSN: 2252-8938, DOI: 10.11591/ijai.v15.i4.pp3683-3691

Hybrid deep learning model for the task scheduling in cloud computing

Kavita Rani[1], Om Prakash Sangwan[1], Ritu Garg[2]
[1]Department of Computer Science and Engineering, Guru Jambheshwar University of Science and Technology, Hisar, India
[2]Department of Computer Engineering, National Institute of Technology Kurukshetra, Thanesar, India

Article Info

Article history:

Received Jun 28, 2025
Revised Jun 10, 2026
Accepted Jul 9, 2026

Keywords:

Cloud computing
Convolutional neural network
Machine learning
Q-learning
Task scheduling

ABSTRACT

Task scheduling plays a crucial role in optimizing performance, reducing costs, and enhancing system reliability by efficiently allocating resources to workloads. Traditional task scheduling methods lack the ability to efficiently manage workloads and resource distribution, leading to potential inefficiencies in performance and energy consumption. To address these limitations, advanced techniques leveraging deep learning and reinforcement learning are explored. This study proposes a deep learning-based model for task scheduling in cloud computing. The model employs a convolutional neural network (CNN) to predict the optimal machines for task allocation. Additionally, Q-learning is integrated with CNN to facilitate load shifting between machines, ensuring efficient utilization of resources. The dataset used in this work consists of task attributes, such as execution time, resource requirements, which were loaded from a CSV file. Comparative analysis with existing models shows that the proposed approach achieves approximately 94% accuracy and consumes less energy than other models, demonstrating its effectiveness in cloud task scheduling.

Corresponding Author:

Kavita Rani
Department of Computer Science and Engineering, Guru Jambheshwar University of Science and Technology
Hisar, India
Email: k.ghanghas67@gmail.com

1. INTRODUCTION

In recent years, cloud computing has become integral to many organizations. It emerged to meet users' demands for accessing computing resources and services on an as-needed basis through internet-driven applications. Cloud computing provides diverse services based on user requirements [1], [2], integrating distributed and parallel computing under a "pay-per-use" model. This eliminates the need for users to purchase software or hardware, only an internet connection is required. Infrastructure as a service (IaaS) is a prominent model that offers virtual servers for data processing and storage [3], [4]. It allows users to run operating systems or applications on rented servers, eliminating hardware maintenance. IaaS also enhances performance and reduces latency by offering server access in nearby locations. Services are governed by quality of service (QoS) parameters and service level agreements (SLAs), with payments based on user-provider agreements [5], [6]. Virtual machines use host resources like RAM and CPU, which can lead to unequal resource distribution. Efficient task scheduling is essential to minimize waiting times and balance resource utilization [7], [8].

Assigning improper virtual machines can cause a high load on any virtual machines and impact system performance. Makespan, cost, and utilization are considered key scheduling factors for determining schedules. Different strategies have been devised to dynamically load-balance tasks. Based on user request, cloud brokers allocate cloud resources efficiently to meet deadlines or SLA requirements [9], [10].

Performance is guaranteed with cloud service provider QoS based checks. Scheduling mechanisms, in accordance to SLAs, are of critical importance to satisfy user expectations [11], [12]. The following shows how hypervisors can support multiple virtual machines on a single hardware layer. These are still a prevalent problem, such as insufficient scheduling and inappropriate task delegation, but all are essential. To ensure processing speed and to prevent imbalance in the system, load balancing is needed [13], [14]. Artificial intelligence (AI) has substantially improved task scheduling with intelligent resource optimization. AI techniques such as machine learning, deep learning, and reinforcement learning predict resource requirements, optimize energy consumption, and adjust resources to accommodate fluctuations in workloads [15], [16]. Heuristic and metaheuristic algorithms such as ant colony optimization (ACO) can be used to search for placement optimizations, minimized idle times, and enhanced energy efficiency. AI also can play a role in multi-cloud practices, improving QoS and scalability. To reduce energy usage and maximize job distribution, a number of AI-based scheduling algorithms have been put forth. While deep learning models may extract intricate task relationships and enhance scheduling policies, machine learning-based approaches use historical workload patterns to forecast resource demand. Using a trial-and-error methodology, reinforcement learning techniques like Q-learning dynamically adjust to shifting workloads. Notwithstanding these developments, current approaches have the drawbacks like static machine learning models struggle with real-time workload fluctuations and fail to generalize well in rapidly changing cloud environments. Deep learning-based models often require extensive training data and computational resources, making them less efficient in real-time scheduling.

The conventional cloud task scheduling approaches suffer from several drawbacks, this research proposes a hybrid deep reinforcement learning method based on convolutional neural network (CNN) and Q-learning to overcome the limitations of the conventional approaches for scheduling tasks in cloud computing environments. The proposed framework uses CNNs to learn patterns of task attributes and resource usage data to derive the features of the tasks and select suitable virtual machines for allocation. To improve the use of the resources without consuming too much energy, adaptive load balancing is incorporated, where optimal task migration strategies are dynamically selected based on the current system state, as determined by the Q-learning. The proposed framework is evaluated against the current scheduling methods by implementing it in Python, and comparing it with the others using performance metrics like accuracy, precision, recall, energy consumption, and task execution time. The results of the experiments show that the hybrid is more accurate at 94%, while reducing energy consumption and increasing the efficiency of the time spent on the task compared to traditional AI-based scheduling.

2. LITERATURE REVIEW

Varma and Bendi [17] proposed a hybrid task scheduling approach that integrates the Al-Biruni earth namib beetle optimization (BENBO) algorithm with bidirectional long short-term memory (Bi-LSTM) network to address multi-objective optimization challenges in cloud computing. In this approach, a combination of scheduling solutions generated by the BENBO optimizer and the Bi-LSTM model is used to generate an optimized task allocation strategy. The hybrid approach successfully reduces the makespan and utilization of the resources. It was tested with respect to important performance criteria such as energy consumption, resource utilization, and makespan. The values obtained in the experimental results were 0.669, 0.535, and 0.381 in terms of energy consumption, resource utilization, and makespan, respectively, showing how effective the proposed scheduling technique was.

Chen and Zheng [18] proposed an AI-based scheduling method for heterogeneous big data network information security. The simulation results indicated its load balancing performance was good in the heterogeneous big data networks' information task scheduling. Moreover, the scheduling method had a great global search and parallel search ability, which was suitable for the large-scale information optimization application in heterogeneous big data networks.

To predict the workload of future tasks, hybrid prediction framework was proposed Lu *et al.* [19], which combines the autoregressive integrated moving average (ARIMA) model with gated recurrent unit (GRU) model. The proposed scheduling strategy was to form multiple priority queues, which will exercise the higher priority task first, and lower priority task second. Moreover, factors like computational resource consumed, model training time, and prediction accuracy were taken into account while formulating the reinforcement learning reward function. As the tasks around the deep learning process reached 'converge' stage, the number of resources being sent to converged or completed tasks was progressively scaled back and reallocated toward other pending tasks in order to optimize the overall resources utilization. The effectiveness of the proposed reinforcement learning-based performance-time optimized (RLPTO) method in scaling-up to large-scale data sets was demonstrated through experimental tests conducted by using more computing nodes, thereby showing good scales and adapts to distributed computing environments.

For the task scheduling algorithm evaluation, Chhabra and Arora [20] came up with a methodology for assigning weights to various performance attributes based on their frequency of occurrence and significance. The former weighted attributes were then combined to calculate priority scores which were in turn used to estimate the performance of each algorithm of the various scheduling algorithms. In addition, the study proposed a machine learning based approach and utilized feature selection techniques to find the most suitable scheduling techniques for cloud computing environments. The findings demonstrated that integrating AI with cloud resource management enhances scheduling efficiency, improves resource utilization, and leads to better overall system performance.

To solve the multi-objective task scheduling problem for independent cloud computing tasks, Kruekaew and Kimpan [21] used the artificial bee colony (ABC) optimization algorithm and the Q-learning reinforcement learning approach. The combination of Q-learning enhanced the exploration/exploitation property of ABC algorithm leading to faster convergence and better scheduling decisions. The proposed multi-objective artificial bee colony algorithm with Q-learning (MOABCQ) algorithm was experimentally tested with a few of the existing scheduling algorithms. The result of the experiments showed that the hybrid method reduced makespan and execution cost, enhanced the workload distribution, increased the throughput and also improved the overall utilization of the resources when compared to the comparative algorithms.

In this regard, Talouki *et al.* [22] proposed a task scheduling algorithm based on a list, which uses an improved task prioritization strategy and task duplication strategies to enhance the scheduling efficiency. The proposed approach uses optimistic cost table downward (OCTd) and optimistic cost table upward (OCTu) procedures to identify priorities of tasks and to create an optimized sequence of execution. The proposed approach has been evaluated against the existing scheduling algorithms and it has been proved that it performs better in terms of various important scheduling metrics such as makespan, speedup, speedup-to-load ratio (SLR), and execution efficiency, which shows its merit in task scheduling optimization.

The provisioning and scheduling of resources in cloud computing environments have increasingly become a challenge due to the growing complexity of these environments, which rely on virtual machines hosted on physical infrastructure in cloud service provider data centers [23]. The authors proposed a framework using the algorithm called quantile regression deep Q-network (QR-DQN) to optimize the performance of resource management and scheduling. This reinforcement learning method is created to learn an optimal policy for long-term resource allocation and scheduling decisions. Results indicated that the proposed framework improved the performance of the task scheduling and improved the efficiency of cloud resource utilization in the computing environment.

Swarup *et al.* [24] discussed the use of cloud computing for various applications such as data analytics, data storage, and internet of things (IoT). The study was centered on task scheduling problem in cloud environment, targeting to minimizing computation resources required to satisfy the resource availability and application deadline constraint. The authors presented a reinforcement learning-based scheduling method, clipped double deep Q-learning (CDDQL) in order to achieve this objective. The approach applied experience replay and a target network to achieve a more stable learning and better scheduling decisions. Experimental results showed that the proposed approach was able to optimally schedule tasks while preserving the resource and deadline constraints.

Alsaidy *et al.* [25] suggested an optimized execution of the particle swarm optimization (PSO) used to support the performance of task scheduling in a cloud computing system. The authors used two heuristic-based initialization methods which are longest job to fastest processor (LJFP) and minimum completion time (MCT) to create good initial particles population to the PSO algorithm. The intention of these initialization procedures was to facilitate the search process and speed up the convergence to optimal scheduling solutions. The proposed LJFP-PSO algorithm and MCT-PSO algorithm have been tested with simulation and compared with the PSO algorithm and some existing task scheduling algorithms. The experimental results showed that both the hybrid methods gave better scheduling performance compared to the benchmark algorithms and were more efficient and effective in terms of overall performance.

3. METHOD

In this research, a hybrid deep learning model integrating CNN and Q-learning is proposed for task scheduling in cloud computing environments. The proposed framework considers resource constraints, dynamic task allocation, and machine utilization to generate efficient scheduling decisions. By combining the feature extraction capability of CNN with the adaptive decision-making ability of Q-learning, the model aims to improve task allocation and resource management. The effectiveness of the proposed scheduling approach is evaluated using key performance metrics, including accuracy, precision, recall, and energy consumption, to assess both scheduling quality and overall system efficiency.

3.1. System model

Work flow diagram, as shown in Figure 1, uses reinforcement learning to optimize task scheduling under resource constraints. It consists of three key parts: the environment, reinforcement learning agent, and evaluation of performance module. Environment is a dynamic task scheduling system with tasks represented by the resource usage, and scheduling status. Based on resource constraints and scheduling efficiency, reinforcement learning agent chooses actions (task assignments) and gets reward/punishment. Q-values are predicted using a deep Q-learning model with a neural network. The agent uses the ϵ-greedy algorithm to choose between exploring and exploiting. The mean squared error (MSE) loss is employed during the training of the network with Adam optimizer. The accuracy, precision, recall, and energy consumption is used to assess performance. The system is designed to manage the allocation of resources efficiently and to reduce amount of resources used.

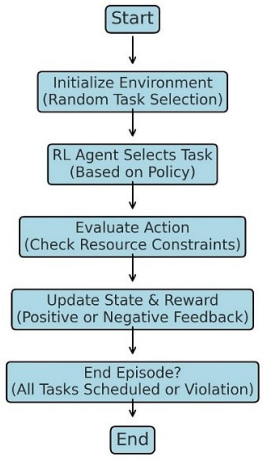

Figure 1. Workflow of the proposed task scheduling approach

3.2. Purposed model

In this proposed methodology, reinforcement learning, along with the CNNs is shown to be an effective combination to solve complex task scheduling problems in cloud computing. This approach combines the ideas of reinforcement learning and CNNs—reinforcement learning for making adaptations of decisions, and CNNs for outstanding feature learning capacity, which is helpful for efficient scheduling in a dynamic cloud environment. One of the greatest strengths of deep learning is its ability to learn end-to-end, meaning that it can learn to extract meaningful characteristics from raw inputs without the need to handcraft them. This feature enables a more reliable scheduling decision and enables the model to be adapted to a broad set of cloud computing applications. 1D-CNNs have recently been shown to attain high performance on processing structured cloud datasets, such as task scheduling, load balancing and so on. CNNs are a type of feed-forward deep neural network that contains a number of convolutional layers which progressively learn to represent features at different levels above which is passed to an output layer that uses the features it has learned to make the final prediction. Figure 2 shows an architecture of the proposed CNN.

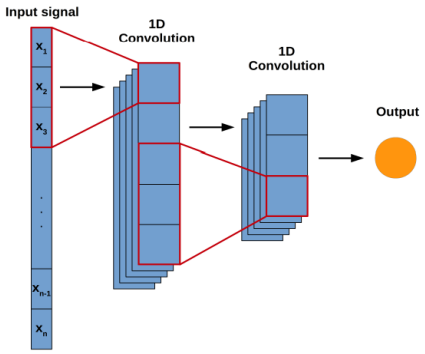

Figure 2. Proposed architecture of CNN

As illustrated in Figure 2, each convolutional stage is composed of a set of learnable convolutional filters followed by a pooling layer. During feature extraction, the input data passes sequentially through these stages, enabling the network to learn increasingly complex and task-specific feature representations. This hierarchical learning process allows the CNN to capture meaningful patterns from cloud computing data, thereby improving the effectiveness of task scheduling. Figure 3 presents the overall architecture of the proposed model along with the distribution of its key parameters.

```
1 model.summary()

Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (1, 10)                   40

 dense_1 (Dense)             (1, 5)                    55

=================================================================
Total params: 95 (380.00 Byte)
Trainable params: 95 (380.00 Byte)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Figure 3. Overall architecture of the proposed task scheduling model

The CNN consists of several important layers such as the input layer, convolutional layer, pooling layer, fully connected layer, and output layer. The input layer processes multidimensional data in the structured format, and features to be passed to the network are often preprocessed to remove dimensionality (normalized) before being fed into the network. This normalization process boosts efficiency in learning and optimizes the model performance. The learnable kernels in the convolutional layer convolve its input feature maps from the previous layer, and then apply a nonlinear activation function to generate a feature map. Each layer produces output by performing a number of convolution operations on the various feature representations of input. The corresponding mathematical formulation of this process is given in (1).

$$y_i^{l+1}(j) = K_i^l * x^l(j) + b_i^l \quad (1)$$

In this information, K_i^l denotes the weights of the ith filter kernel at layerl, while b_i^l denotes the bias of the ith filter kernel at layer l, $\mathrm{x}^l(j)$ refers to the jth local region at layer l, and $y_i^{l+1}(j)$ represents the input of the jth neuron in frame i of layer "l+1". The symbol * represents the operation of dot product between the convolutional kernel and the local receptive field it is used with. Once the convolution process is finished, an activation function is applied to the generated feature maps to add non-linearity into the feature maps. When this step is applied, the multi-dimensional features extracted from the data may not be linearly separable in the original feature space, but they become more linearly separable in the higher-dimensional space. The rectified linear unit (ReLU) activation function is used in this study. It has several of its merits, one being that its gradient is fixed (equal to 1) for the positive input values, so as to offset the vanishing gradient problem in training. The mathematical description of the ReLU function is given as in (2).

$$a_i^{l+1}(j) = f\left(y_i^{l+1}(j)\right) = max\{0, y_i^{l+1}(j)\} \quad (2)$$

In this context $y_i^{l+1}(j)$ represents the output value obtaind after convolution operation, while $a_i^{l+1}(j)$ shows the activation value of $y_i^{l+1}(j)$. The aim of the pooling layer is to decrease the dimensionality of the feature maps by downsampling feature maps that are made up of large input matrices to smaller feature maps. This significantly diminishes the amount of parameters to be learned and decreases the computational complexity and also helps prevent overfitting the model. Max pooling is one of the most popular pooling methods, which picks out the largest value in each receptive field. Math definition of max pooling is defined in (3).

$$P_i^{l+1}(j) = \max_{(j-1)W+1 \le t \le jW} \{q_i^l(t)\} \quad (3)$$

Here, $q_i^l(t)$ represents the tth neuron's value in the ith feature at l layer, $(t \in [(j-1)W+1, jW])$; W is the pooling region's width. Also, the neuron's value at layer "l+1" is $P_i^{l+1}(j)$. After the last pooling layer, there is a fully connected layer. This last pooling stage generates feature maps that are flattened into a one dimensional vector and that's used as input to the fully connected layer. This layer has each input neuron linked to all the

output neurons, enabling the network to integrate the local pattern of features captured by the convolutional and pooling layers into a single, global representation. The integrated representation is then employed for the final classification or prediction. The fully connected layer can be mathematically described as (4).

$$z^{l+1}(j) = f\left(\sum_{i=1}^{m}\sum_{t=1}^{n} W_{itj}^{l}\, a_i^l(t) + b_j^l\right) \quad (4)$$

Where W_{itj}^{l} is the weight connecting the tth neuron in the ith feature at layer l to the jth neuron at layer "l+1". The logit value corresponding to the jth neuron at layer "l+1" is $z^{l+1}(j)$. Moreover, the network offset is b_j^l while $a_i^l(t)$ is the tth neuron's output value in the ith feature at the preceding layer l. Finally, $f(\cdot)$ is the activation function ReLU. A softmax classifier is generally employed in the output layer to generate the final class label predictions. The softmax function is an extension of logistic regression to multi-class problems, normalizing raw scores from the logistic regression output to produce normalized probability distributions. This property makes it useful in multi-class classification problems for assigning each input sample to one of the different classes according to the highest probability, as defined in (5).

$$Q(j) = softmax\left(z^o(j)\right) = \frac{e^{z^o(j)}}{\sum_{k=1}^{M} e^{z^o(k)}} \quad (5)$$

Here $z^o(j)$ refers to the logit produced by the j-th neuron in the output layer and M is the number of target classes. The CNN is a type of neural network that processes information in a sequence or a vector of data, which is referred to as a 1D-CNN. Hence, one-dimensional convolutional filters are used in the network to capture the meaningful patterns, along a single dimension. As a result, the convolutional and pooling layers generate one-dimensional feature representations and these are then applied to classification or prediction.

4. RESULTS AND DISCUSSION

In this study, the task scheduling for cloud computing with machine learning techniques is discussed. The proposed framework uses a CNN to perform machine level prediction and the method of Q-learning to help shift the load dynamically between machines. The effectiveness of the proposed approach is assessed against an established approach based on a number of metrics. For this, the accuracy, precision, recall, and energy consumption are taken into account to evaluate the performance of the scheduling model. The results of performance analysis of the proposed model are shown in Table 1.

Table 1. Performance analysis of task scheduling methods

Model	Accuracy (%)	Precision (%)	Recall (%)
Neural networks [26]	84.10	84	84
Dynamic resource allocation [15]	86.10	86	86
Proposed	92.34	92	92

The proposed approach was tested in terms of its effectiveness against the existing task scheduling methods like neural network-based models and dynamic resource allocation methods as shown in Figure 4. When comparing the proposed model with the existing models, the accuracy, precision, and recall values are consistently higher with each performance metric being above 90%. The results from the aforementioned, clearly surpass the benchmark methods. The improved performance may be due to the model's ability to represent the complex characteristics of tasks and to process large-scale, time-dependent data. Conventional neural network-based approaches and dynamic resource allocation methods, however, tend to work well in simpler settings but have drawbacks of scalability and prediction in larger and more complex deployment of cloud workloads. Thus, the proposed framework is more appropriate for large-scale cloud computing environments where real-time scheduling decisions must be taken and efficient resource management is desired. Moreover, the achieved improvement demonstrates the benefits of combining deep learning and reinforcement learning to enhance task scheduling performance. Overall, the performance of the model presented in Figure 4 is satisfactory, confirming the robustness and efficiency of the proposed model for various evaluation metrics. The results of the energy efficiency analysis of the neural networks [26] is 11.2 joules, dynamic resource allocation [15] is 9.5 joules, and proposed framework is 6.5 joules.

As it is shown in Figure 5, the energy efficiency of the proposed algorithm is compared with existing algorithms like neural networks and dynamic resource allocation. The analysis shows that the proposed algorithm is the most energy saving algorithm compared to other algorithms. In cloud environments where energy consumption is supposed to be desired, consuming too much energy causes operational costs to increase and impacts to the environment. The suggested model keeps the resource contention as low as

possible and energy usage as low as possible as well. As shown in Figure 5, the proposed method results in the lowest energy consumption (6.5 joules), in comparison with the traditional methods. This substantial cut is due to better task prediction and the more effective use of resources. In order to encourage sustainable usage of cloud computing it is crucial to introduce methods that save energy in the scheduling procedure. The energy-saving feature of the proposed model results in it being a potential solution design for implementation of this kind of cloud application in real life, for which energy is a critical factor.

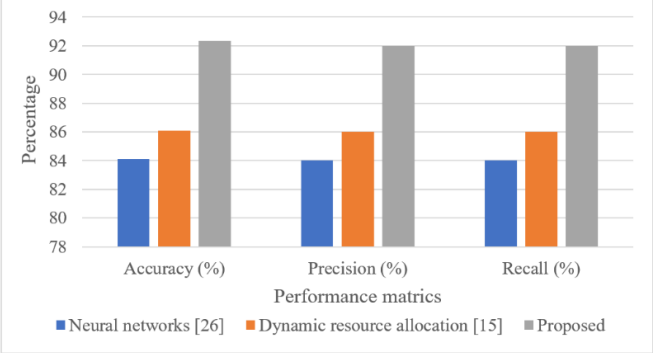

Figure 4. Performance comparison of task scheduling methods

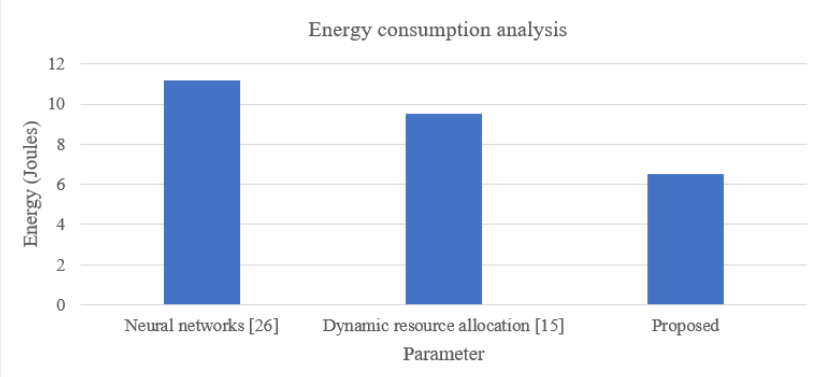

Figure 5. Energy efficiency comparison of task scheduling methods

5. CONCLUSION

MapReduce is a popular programming paradigm for the parallel processing of massive amounts of data in a cloud-based computing environment, developed by Google. It has two main phases: map and reduce. During the map phase, the large sets of user data are broken into smaller sub-tasks and then they are scheduled and assigned to the available cloud resources in an appropriate way, using appropriate scheduling strategies. The intermediate results are sent to the reduce phase after they are produced on the distributed computing nodes, and finally the result is sent to the user. This way, it becomes easier to perform a lot of complex processing tasks, and it also makes the process faster. This study aims to propose a deep learningbased model for task scheduling in cloud computing and implement it in python. The effectiveness of the proposed method is compared to the existing methods like neural networks and dynamic resource allocation techniques. Experimental results show that the proposed model shows accuracy, precision, and recall greater than 92%, while comparative models show approximately 8% less.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Kavita Rani	✓	✓	✓	✓	✓	✓		✓	✓	✓			✓	
Om Prakash Sangwan	✓	✓		✓		✓		✓	✓	✓	✓	✓		
Ritu Garg	✓	✓				✓		✓	✓	✓	✓	✓		

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

Task data was loaded from a CSV file and randomized to simulate different scenarios.

REFERENCES

[1] J. Xue, L. Li, S. Zhao, and L. Jiao, "A study of task scheduling based on differential evolution algorithm in cloud computing," in *2014 International Conference on Computational Intelligence and Communication Networks*, Nov. 2014, pp. 637–640, doi: 10.1109/CICN.2014.142.

[2] V. Ahari, R. Venkatesan, and D. P. P. Latha, "A survey on task scheduling using intelligent water drops algorithm in cloud computing," in *2019 3rd International Conference on Trends in Electronics and Informatics*, Apr. 2019, pp. 39–45, doi: 10.1109/ICOEI.2019.8862777.

[3] S. Mittal and A. Katal, "An optimized task scheduling algorithm in cloud computing," in *2016 IEEE 6th International Conference on Advanced Computing*, Feb. 2016, pp. 197–202, doi: 10.1109/IACC.2016.45.

[4] X. Chen *et al.*, "A WOA-based optimization approach for task scheduling in cloud computing systems," *IEEE Systems Journal*, vol. 14, no. 3, pp. 3117–3128, Sep. 2020, doi: 10.1109/JSYST.2019.2960088.

[5] T. Aladwani, "Improving tasks scheduling performance in cloud computing environment by using analytic hierarchy process model," in *2017 International Conference on Green Informatics*, Aug. 2017, pp. 98–104, doi: 10.1109/ICGI.2017.13.

[6] M. S. Sanaj and P. M. Prathap, "An enhanced round robin (ERR) algorithm for effective and efficient task scheduling in cloud environment," in *2020 Advanced Computing and Communication Technologies for High Performance Applications*, Jul. 2020, pp. 107–110, doi: 10.1109/ACCTHPA49271.2020.9213198.

[7] X. J. Wei, W. Bei, and L. Jun, "SAMPGA task scheduling algorithm in cloud computing," in *2017 36th Chinese Control Conference*, Jul. 2017, pp. 5633–5637, doi: 10.23919/ChiCC.2017.8028252.

[8] J. Kaur and B. K. Sidhu, "A new flower pollination based task scheduling algorithm in cloud environment," in *2017 4th International Conference on Signal Processing, Computing and Control*, Sep. 2017, pp. 457–462, doi: 10.1109/ISPCC.2017.8269722.

[9] S. Srichandan, T. A. Kumar, and S. Bibhudatta, "Task scheduling for cloud computing using multi-objective hybrid bacteria foraging algorithm," *Future Computing and Informatics Journal*, vol. 3, no. 2, pp. 210–230, Dec. 2018, doi: 10.1016/j.fcij.2018.03.004.

[10] N. Mansouri, B. M. H. Zade, and M. M. Javidi, "Hybrid task scheduling strategy for cloud computing by modified particle swarm optimization and fuzzy theory," *Computers & Industrial Engineering*, vol. 130, pp. 597–633, Apr. 2019, doi: 10.1016/j.cie.2019.03.006.

[11] S. Varshney and S. Singh, "An optimal bi-objective particle swarm optimization algorithm for task scheduling in cloud computing," in *2018 2nd International Conference on Trends in Electronics and Informatics*, May 2018, pp. 780–784, doi: 10.1109/ICOEI.2018.8553728.

[12] G. Geetha, A. Khare, K. Mahawar, and B. Bajaj, "Strategic load balancing: game theory approach for optimized resource allocation in cloud computing," in *2024 8th International Conference on Inventive Systems and Control*, Jul. 2024, pp. 345–350, doi: 10.1109/ICISC62624.2024.00066.

[13] N. K. Rajpoot, P. Singh, and B. Pant, "Nature-inspired load balancing algorithms for resource allocation in cloud computing," in *2023 International Conference on Computational Intelligence and Sustainable Engineering Solutions*, Apr. 2023, pp. 827–832, doi: 10.1109/CISES58720.2023.10183630.

[14] A. Thakur and M. S. Goraya, "RAFL: a hybrid metaheuristic based resource allocation framework for load balancing in cloud computing environment," *Simulation Modelling Practice and Theory*, vol. 116, Apr. 2022, doi: 10.1016/j.simpat.2021.102485.

[15] S. Chhabra and A. K. Singh, "Dynamic resource allocation method for load balance scheduling over cloud data center networks," *Journal of Web Engineering*, vol. 20, no. 8, pp. 2269–2284, Nov. 2021, doi: 10.13052/jwe1540-9589.2083.

[16] S. Swarnakar, R. Kumar, S. Krishn, and C. Banerjee, "Improved dynamic load balancing approach in cloud computing," in *2020 IEEE 1st International Conference for Convergence in Engineering*, Sep. 2020, pp. 195–199, doi: 10.1109/ICCE50343.2020.9290602.

[17] P. J. Varma and S. R. Bendi, "Multiobjective hybrid al-biruni earth namib beetle optimization and deep learning based task scheduling in cloud computing," *Sustainable Computing: Informatics and Systems*, vol. 44, Dec. 2024, doi: 10.1016/j.suscom.2024.101053.

[18] Y. Chen and Y. Zheng, "Multi-task scheduling method for information security of heterogeneous big data networks based on artificial intelligence technology," in *2024 IEEE 2nd International Conference on Sensors, Electronics and Computer Engineering*, Aug. 2024, pp. 53–58, doi: 10.1109/ICSECE61636.2024.10729453.

[19] X. Lu, C. Liu, S. Zhu, Y. Mao, P. Lio, and P. Hui, "RLPTO: a reinforcement learning-based performance-time optimized task and resource scheduling mechanism for distributed machine learning," *IEEE Transactions on Parallel and Distributed Systems*, vol. 34, no. 12, pp. 3266–3279, Dec. 2023, doi: 10.1109/TPDS.2023.3317388.
[20] M. Chhabra and G. Arora, "The impact of machine learning (ML) driven algorithm ranking and visualization on task scheduling in cloud computing," in *2023 3rd International Conference on Advancement in Electronics & Communication Engineering*, Nov. 2023, pp. 969–972, doi: 10.1109/AECE59614.2023.10428475.
[21] B. Kruekaew and W. Kimpan, "Multi-objective task scheduling optimization for load balancing in cloud computing environment using hybrid artificial bee colony algorithm with reinforcement learning," *IEEE Access*, vol. 10, pp. 17803–17818, 2022, doi: 10.1109/ACCESS.2022.3149955.
[22] R. Talouki, M. H. Shirvani, and H. Motameni, "A heuristic-based task scheduling algorithm for scientific workflows in heterogeneous cloud computing platforms," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 8, pp. 4902–4913, Sep. 2022, doi: 10.1016/j.jksuci.2021.05.011.
[23] T. Oudaa, H. Gharsellaoui, and S. B. Ahmed, "An agent-based model for resource provisioning and task scheduling in cloud computing using DRL," *Procedia Computer Science*, vol. 192, pp. 3795–3804, 2021, doi: 10.1016/j.procs.2021.09.154.
[24] S. Swarup, E. M. Shakshuki, and A. Yasar, "Task scheduling in cloud using deep reinforcement learning," *Procedia Computer Science*, vol. 184, pp. 42–51, 2021, doi: 10.1016/j.procs.2021.03.016.
[25] S. A. Alsaidy, A. D. Abbood, and M. A. Sahib, "Heuristic initialization of PSO task scheduling algorithm in cloud computing," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 6, pp. 2370–2382, Jun. 2022, doi: 10.1016/j.jksuci.2020.11.002.
[26] 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, doi: 10.1016/j.iotcps.2024.02.002.

BIOGRAPHIES OF AUTHORS

Kavita Rani holds a B.Tech. and M.Tech. in Computer Science and Engineering. She is currently a Ph.D. student in Computer Science and Engineering at Guru Jambheshwar University of Science and Technology, Hisar, Haryana, India. She is focusing her research on energy efficiency in cloud computing. She can be contacted at email: k.ghanghas67@gmail.com.

Om Prakash Sangwan is presently working as a professor and chairman, Department of Computer Science and Engineering, Guru Jambheshwar University of Science and Technology, Hisar, Haryana, India. He is also having an additional charge as dean, international affairs, Guru Jambheshwar University of Science and Technology w.e.f 24th October 2025. He received his Ph.D. in Computer Science and Engineering and Master of Technology (M.Tech.) degree in Computer Science and Engineering with distinction in research work from Guru Jambheshwar University of Science and Technology, Hisar, Haryana. He is also CISCO Certified Network Associate (CCNA) and CISCO Certified Academic Instructor (CCAI). He is having 21 years of experience. His area of research is software engineering, AI, and machine learning. More than 150 publications in international/national journals, and conferences in his credit with the highest impact factor of 6.0. Earlier to this, he worked as dy. director with Amity Resource Centre for Information Technology (ARCIT), and LMCand head, CISCO Regional Networking Academy, Amity Institute of Information Technology, Amity University, Uttar Pradesh. 8 research scholars have been awarded Ph.D. degrees under his supervision and 6 research scholars are pursuing Ph.D. under his guidance. He can be contacted at email: sangwan0863@gmail.com.

Ritu Garg is an associate professor in the Department of Computer Engineering at the National Institute of Technology Kurukshetra, Thanesar, India. She holds a B.Tech. and M.Tech. in Computer Science and Engineering, and earned her Ph.D. from the National Institute of Technology Kurukshetra, Thanesar, India, with over 20 years of experience in teaching and research. She has authored 86 publications, including 32 papers in SCI/Scopus indexed journals, numerous book chapters, and conference papers. As co-PI, she completed a TEQIP-III project on brain tumor detection using MRI images. She has supervised five Ph.D. scholars and 27 M.Tech. dissertations, and contributed extensively to curriculum design, NBA accreditation, and institutional leadership roles. She is currently working on MeitY GOI sponsored proof-of-concept project titled "Crop health monitoring and yield prediction using deep learning with drones imagery" and another MeitY GOI sponsored project entitled "Capacity building for human resource development in unmanned aircraft system (drone and related technologies)" under drone applications work theme worth 2.5 Cr. She can be contacted at email: ritu.59@gmail.com.