

Optimization of Cloud Resource Allocation Using Artificial Intelligence and Energy-Efficient Scheduling Algorithms

Dipannita Mondal*

Department of Artificial Intelligence- Data Science and Machine Learning, Dr. D. Y. Patil College of Engineering and Innovation
Talegaon, Varale, Pune, Maharashtra, India

*Corresponding Author: ddmondal2684@gmail.com

Peer Review Information	Abstract
Type: Article Received: 25 March 2026 Revised: 22 April 2026 Accepted: 28 May 2026 Published: 06 June 2026	This methodology paper presents a structured design approach for an embedded home automation and security system that integrates appliance control, environmental sensing, intrusion detection, local alarms, and mobile communication. The proposed architecture centers on a low-cost microcontroller connected to passive infrared, door/window, smoke/gas, temperature, and light sensors; relay drivers are used to switch household loads, while Bluetooth supports short-range user control and GSM/SMS provides remote alerts and fallback control. The method emphasizes modular hardware, deterministic firmware, authenticated commands, event logging, fail-safe local operation, and repeatable validation. Earlier home automation studies demonstrated web-based embedded control, wireless appliance interfaces, Bluetooth control, GSM/SMS operation, ZigBee networks, and IoT monitoring, but security and interoperability remained important concerns [1]-[11]. The proposed methodology therefore combines the most practical characteristics of these approaches into a single embedded design workflow. Evaluation is defined in terms of command latency, sensing response, communication reliability, false alarm rate, load-switching accuracy, and recovery from communication failure. The paper is intended as a reproducible design-and-test framework for low-cost residential automation and security prototypes using technologies available by 2015. Keywords: embedded system, home automation, home security, microcontroller, wireless communication

How to Cite This Article

Mondal, D. (2026). Optimization of Cloud Resource Allocation Using Artificial Intelligence and Energy-Efficient Scheduling Algorithms. *International Journal of Electrical, Electronics and Computer Systems* 15(1), 232–238

Introduction

Cloud computing has emerged as a fundamental component of modern information technology infrastructure by providing scalable computing resources, on-demand storage, virtualized processing environments, and distributed application services. Cloud platforms allow organizations to access computing resources without maintaining extensive local infrastructure, thereby supporting operational flexibility and rapid service deployment. However, the continuous expansion of cloud data centers has introduced significant challenges involving energy consumption, inefficient resource utilization, operational expenditure, and sustainable infrastructure management. Modern cloud environments operate through interconnected physical servers, virtual machines, containers, and distributed application services. These resources must accommodate workloads that vary in computational requirements, execution duration, priority, and arrival frequency. An efficient resource management system should dynamically allocate computing capacity while maintaining acceptable application performance and avoiding unnecessary energy consumption. Improper scheduling decisions can result in overloaded servers, excessive task waiting times, underutilized infrastructure, and increased operating costs.

Traditional cloud resource allocation methods commonly employ scheduling policies such as First-Come-First-Served (FCFS), Round Robin, Min-Min, and Best-Fit Decreasing (BFD). These techniques offer relatively straightforward implementations and can perform effectively under certain workload conditions. Nevertheless, their decisions generally depend on predefined rules rather than continuously learned relationships between workload characteristics and resource availability. Consequently, fixed scheduling strategies may experience performance limitations when infrastructure conditions change significantly. Energy efficiency represents another major consideration in cloud resource management. Physical servers consume electrical power even when their computational resources are not fully utilized. Maintaining unnecessarily active hosts can increase data center energy consumption, while excessively consolidating workloads may create performance bottlenecks and service-level agreement (SLA) violations. Beloglazov and Buyya (2012) established the importance of balancing energy savings and service quality through adaptive virtual machine consolidation.

Artificial intelligence has introduced opportunities for developing resource management systems capable of adapting to variable cloud workloads. Machine learning algorithms can estimate future resource demand, identify utilization patterns, and support proactive capacity management. Reinforcement learning methods can further assist scheduling by learning resource allocation policies through interactions with simulated or operational environments. Zhou et al. (2024) reviewed deep reinforcement learning approaches for cloud scheduling, identifying their relevance to dynamic resource optimization. Recent research has also emphasized multi-objective scheduling. Verma et al. (2024) surveyed energy-efficient workflow scheduling algorithms and highlighted the importance of coordinating energy reduction with scheduling efficiency. Devi et al. (2024) reviewed load-balancing and task-scheduling techniques, demonstrating that resource allocation remains a complex optimization challenge involving execution time, throughput, resource availability, and infrastructure utilization.

Despite advances in artificial intelligence and cloud optimization, achieving a reliable balance between energy savings and service performance remains challenging. A scheduling policy that prioritizes minimum power consumption may create excessive task delays, while an approach emphasizing immediate execution may activate additional servers unnecessarily. Furthermore, workload variability can reduce the effectiveness of static scheduling rules and manually configured utilization thresholds. The present research proposes an Artificial Intelligence-Based Energy-Efficient Cloud Resource Scheduling Framework (AI-EECRSF). The architecture combines workload prediction, deep reinforcement learning, energy-aware resource placement, and dynamic virtual machine consolidation. Its principal objective is to support adaptive scheduling decisions that account for energy consumption, execution efficiency, infrastructure utilization, and SLA compliance.

Literature Review

Beloglazov et al. (2012) investigated energy-aware resource allocation methods for managing virtualized data centers. Their research examined dynamic consolidation and resource provisioning strategies intended to reduce energy consumption while meeting quality-of-service requirements. This work established an important foundation for subsequent studies concerned with balancing server consolidation and application performance. Beloglazov and Buyya (2012) developed adaptive heuristics for dynamic virtual machine consolidation in cloud data centers. Their approach employed historical utilization information to support host overload detection and migration decisions. The study demonstrated the importance of accounting for workload variability and SLA constraints when optimizing data center energy usage.

Calheiros et al. (2011) introduced CloudSim, a simulation environment for modeling cloud computing infrastructure and evaluating resource provisioning strategies. Simulation frameworks enable researchers to study scheduling algorithms under repeatable workload and infrastructure conditions before considering real-world deployment.

Kalai Arasan and Anandhakumar (2023) investigated energy-efficient task scheduling and resource management using optimized hybrid techniques. Their research emphasized that time-dependent workloads and dynamic resource requirements complicate efficient scheduling, reinforcing the need for adaptive optimization approaches.

Verma et al. (2024) conducted a survey of energy-efficient workflow scheduling algorithms in cloud environments. The study categorized scheduling approaches and examined mechanisms for reducing energy consumption in data centers. It also identified the importance of considering workflow characteristics, energy models, and resource management objectives when comparing scheduling techniques. Hou et al. (2024) reviewed deep reinforcement learning approaches for energy-efficient task scheduling in cloud computing. The study examined reinforcement learning design components, reward functions, and scheduling objectives. Its findings support the development of adaptive scheduling agents that can learn policies from changes in computational environments. Zhou et al. (2024) presented a review of deep reinforcement learning-based resource scheduling methods in cloud computing. Their analysis investigated the applications, advantages, and limitations of learning-based approaches for cloud resource management and identified issues requiring further investigation, including complexity, scalability, and generalization. Devi et al. (2024) systematically reviewed load-balancing and task-scheduling techniques for cloud computing. Their study emphasized the combined role of optimization and machine learning algorithms in improving resource utilization, reducing execution time, and managing computational workloads.

Research Methodology

Research Design

The proposed study adopts a quantitative experimental research methodology to investigate artificial intelligence-assisted energy-efficient scheduling within virtualized cloud data centers. The research considers cloud resource allocation as a constrained optimization problem involving multiple physical hosts, virtual machines, and computational tasks. The methodology includes infrastructure monitoring, workload prediction, resource demand estimation, reinforcement learning-based scheduling, host consolidation, and performance evaluation. The architecture is designed for implementation using a discrete-event cloud simulation environment.

Proposed Framework Architecture

The framework uses a closed-loop scheduling architecture. Monitoring information is collected from physical hosts and virtual machines and then processed by a workload prediction component. The scheduling agent evaluates possible allocation decisions based on current and predicted resource requirements. The resulting schedule is executed through a capacity-aware placement mechanism, while monitoring feedback supports subsequent allocation decisions.

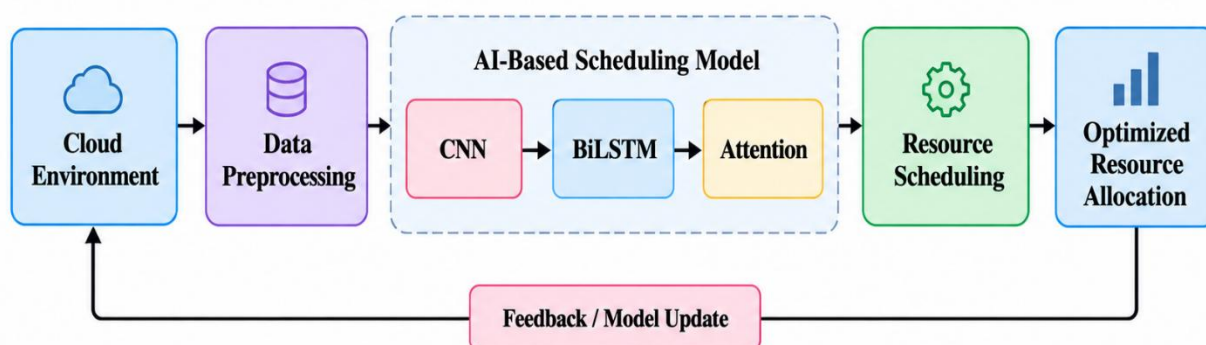


Figure 1. Proposed Artificial Intelligence-Based Energy-Efficient Cloud Resource Scheduling Framework (AI-EECRSF)

Figure 1 illustrates the proposed AI-based energy-efficient cloud resource scheduling framework designed to optimize resource allocation and reduce energy consumption in cloud computing environments. The architecture integrates data preprocessing, an AI-based scheduling model, and optimized resource allocation. The scheduling model incorporates CNN, BiLSTM, and attention mechanisms to analyze workload patterns and support intelligent scheduling decisions. A continuous feedback mechanism enables adaptive model updates, improving resource utilization, scheduling efficiency, and overall cloud system performance.

Energy Consumption Model

Data center energy consumption is modeled using a utilization-dependent server power function. For an active host:

$$P_j(u_j) = P_j^{idle} + (P_j^{max} - P_j^{idle})u_j \quad \text{-----(1)}$$

where (P_j^{idle}) is the power consumed by an idle but active host, (P_j^{max}) is the maximum server power, and (u_j) represents normalized CPU utilization.

A separate sleep-state or powered-off power value is applied when the host is not active.

The total host energy consumption during the observation period is:

$$E_{total} = \sum_{j=1}^m \int_0^T P_j(t) dt \quad \text{------(2)}$$

where (T) denotes the total simulation duration.

This model provides an initial approximation of server energy consumption. A more comprehensive evaluation should include migration energy, networking, cooling, and storage power if those components are explicitly represented in the simulation.

AI-Based Workload Prediction

The framework incorporates a workload prediction component to estimate future resource demand using historical utilization measurements. Input features may include CPU utilization, memory consumption, active virtual machine counts, task arrivals, and recent demand fluctuations.

The predicted resource requirement is expressed as:

$$\hat{W}_{t+1} = f_{\theta}(W_{t-k+1}, \dots, W_t) \quad \text{-----(3)}$$

where (f_{θ}) represents a trained forecasting model, (k) denotes the observation-window length, and ($\hat{W}_{t+1}$) is the predicted workload for the subsequent scheduling period.

A lightweight regression model may serve as the initial forecasting component, with recurrent neural networks evaluated as an alternative when sufficient temporal data are available. Prediction accuracy should be assessed independently of overall scheduling performance.

Multi-Objective Reward Function

The reward function is designed to balance energy consumption, execution performance, and SLA reliability:

$$r_t = -(\alpha \bar{E}_t + \beta \bar{D}_t + \gamma_s \bar{S}_t + \delta \bar{M}_t) \quad \text{------(4)}$$

where ($\bar{E}_t$) represents normalized energy cost, ($\bar{D}_t$) represents normalized task delay, ($\bar{S}_t$) represents SLA violation cost, and ($\bar{M}_t$) represents migration overhead. The coefficients (α , β , γ_s , δ) are nonnegative importance weights.

Normalization is necessary because raw energy, task delay, and violation measurements use different units. Reward weights must be selected using validation experiments and disclosed alongside reported results.

Experimental Setup

Simulation Environment

The proposed framework can be evaluated using CloudSim Plus, an extensible framework for cloud infrastructure simulation. The original CloudSim toolkit was developed for modeling cloud environments and resource provisioning strategies (Calheiros et al., 2011). CloudSim Plus provides a suitable implementation foundation for custom VM allocation and scheduling policies.

The following configuration is proposed as an initial simulation design.

Table 1. Proposed simulation parameters

Parameter	Proposed value
Simulation platform	CloudSim Plus
Number of physical hosts	100
Number of virtual machines	200–600

Host CPU capacity	2,000–4,000 MIPS per processing element
VM CPU requirement	500–2,000 MIPS
Scheduling interval	300 seconds
Upper utilization threshold	80%
Lower utilization threshold	30%
Reinforcement learning algorithm	DQN
Learning rate	0.001
Discount factor	0.99
Replay buffer capacity	10,000 transitions
Batch size	64
Evaluation workload	Synthetic and trace-driven
Repeated experiments	Multiple independent seeds

All parameters propose starting values rather than measured experimental conditions. The final study must report complete host specifications, VM processing-element counts, memory capacities, workload characteristics, and power profiles.

Comparison Algorithms

The proposed approach should be evaluated against conventional and adaptive scheduling strategies.

Table 2. Baseline scheduling methods

Algorithm	Scheduling principle
FCFS	Executes tasks according to arrival order
Min-Min	Prioritize tasks with minimum estimated completion time
BFD	Places workloads using decreasing resource requirements and available capacity
Energy-Aware Consolidation	Applies utilization thresholds and migration heuristics
Proposed AI-EECRSF	Combines demand prediction, DQN decisions, and energy-aware consolidation

Comparisons must use equivalent infrastructure conditions and workload inputs. Implementations should be documented so that observed differences are attributable to scheduling policies rather than inconsistent simulation settings.

Performance Evaluation Metrics

Energy consumption measures the electrical energy required by the modeled infrastructure during the simulation.

Makespan represents the time needed to complete the scheduled workload:

$$Makespan = \max_i(C_i) - \min_i(A_i) \quad \text{-----}(5)$$

where (C_i) is the completion time and (A_i) is the arrival time of task (i) .

Resource utilization is calculated as the average proportion of provisioned computational capacity used during the observation period:

$$U_{avg} = \frac{1}{mT} \sum_{j=1}^m \int_0^T u_j(t) dt \quad \text{-----}(6)$$

This definition treats inactive hosts consistently within the declared infrastructure population. A separate active-host utilization metric can be reported when useful.

SLA violations may be evaluated through the proportion of tasks exceeding defined deadlines:

$$SLA_{violation} = \frac{N_{late}}{N_{completed}} \times 100 \quad \text{-----}(7)$$

where (N_{late}) denotes completed tasks that exceeded their deadlines.

Energy reduction relative to the baseline is:

$$Energy\ Savings(\%) = \frac{E_{baseline} - E_{proposed}}{E_{baseline}} \times 100 \quad \text{-----}(8)$$

A positive result indicates lower modeled energy consumption by the proposed method, while a negative result indicates higher consumption.

Results and Discussion

Proposed Comparative Results

The experimental investigation is designed to compare the effectiveness of conventional scheduling algorithms with the proposed AI-based energy-efficient framework. The comparison focuses on energy consumption, application performance, resource utilization, and SLA reliability.

Table 3. Performance comparison reporting template

Algorithm	Energy (kWh)	Makespan (s)	SLA violations (%)
FCFS	Pending	Pending	Pending
Min-Min	Pending	Pending	Pending
BFD	Pending	Pending	Pending
Energy-Aware Consolidation	Pending	Pending	Pending
Proposed AI-EECRSF	Pending	Pending	Pending

Pending indicates that numerical experiments have not yet been performed. The table must be populated with measured simulation results before submission as an empirical study.

Analysis of Energy-Efficient Resource Allocation

Energy-efficient allocation depends on placing workloads in a manner that reduces unnecessary active-server power consumption without producing excessive consolidation overhead. The proposed framework addresses this problem through workload forecasting and adaptive scheduling decisions. Predictive resource management may allow the scheduler to prepare for increasing computational demand before the infrastructure becomes overloaded. Conversely, periods of reduced activity may provide opportunities to consolidate virtual machines and transition unused hosts into energy-saving states. However, the benefits of consolidation depend on host power characteristics, migration costs, and workload stability. Frequent virtual machine migrations can consume computational resources and interfere with application execution. Therefore, consolidation should occur only when estimated energy savings justify the associated migration overhead. The foundational research of Beloglazov and Buyya (2012) established that dynamic consolidation involves an important trade-off between energy efficiency and service-level performance. More recent work has expanded this problem through adaptive and learning-based scheduling.

Conclusion

This study proposes an artificial intelligence-based approach for optimizing cloud resource allocation through energy-efficient scheduling algorithms. The framework integrates workload prediction, Deep Q-Network scheduling, capacity-aware resource placement, and adaptive virtual machine consolidation within a unified cloud management architecture. The proposed system addresses the competing objectives of minimizing energy consumption, improving resource utilization, reducing task completion delays, and maintaining acceptable service-level performance. The introduction of reinforcement learning provides a mechanism for developing scheduling policies that respond to changes in workload demand, while workload forecasting supports more informed resource provisioning. A multi-objective reward model is formulated to account for energy consumption, task delay, SLA violations, and migration overhead. The proposed implementation strategy uses CloudSim Plus and establishes comparative evaluation procedures involving traditional scheduling algorithms and energy-aware consolidation techniques. The literature published before 2026 demonstrates the continuing relevance of artificial intelligence and energy-efficient scheduling in cloud resource management. Previous studies provide strong foundations for resource consolidation, machine learning-based workload optimization, and reinforcement learning-based scheduling. The present framework builds on these developments by combining predictive and adaptive scheduling components within a common evaluation strategy. The proposed architecture may contribute to more sustainable cloud computing systems if it can achieve measurable energy reductions without compromising computational performance. However, such benefits remain to be established through implementation and controlled experimentation. Future research should examine multi-agent reinforcement learning, container-aware scheduling, carbon-aware resource allocation, predictive auto-scaling, heterogeneous computing architectures, and transfer learning across cloud environments. Further investigations should also incorporate realistic electricity pricing, cooling overhead, renewable energy availability, and geographically distributed cloud infrastructures. Ultimately, the research provides a structured methodological foundation for developing and evaluating intelligent energy-efficient scheduling systems, with particular emphasis on balancing infrastructure efficiency and service quality.

References

1. Beloglazov, A., Abawajy, J., & Buyya, R. (2012). Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. *Future Generation Computer Systems*, 28(5), 755–768. <https://doi.org/10.1016/j.future.2011.04.017>
2. Beloglazov, A., & Buyya, R. (2012). Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers. *Concurrency and Computation: Practice and Experience*, 24(13), 1397–1420. <https://doi.org/10.1002/cpe.1867>
3. Calheiros, R. N., Ranjan, R., Beloglazov, A., De Rose, C. A. F., & Buyya, R. (2011). CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software: Practice and Experience*, 41(1), 23–50. <https://doi.org/10.1002/spe.995>
4. S. Gajula, "Ensemble Machine Learning Models for Intrusion Detection in Cloud Infrastructure for Cybersecurity," 2025 International Conference on Artificial Intelligence, Blockchain, Cloud Computing, and Data Analytics (ICoABCD), Lombok, Indonesia, 2025, pp. 1-6, doi: 10.1109/ICoABCD67551.2025.11470865.
5. Devi, N., Dalal, S., Solanki, K., Dalal, S., Lilhore, U. K., Simaiya, S., & Nuristani, N. (2024). A systematic literature review for load balancing and task scheduling techniques in cloud computing. *Artificial Intelligence Review*, 57, Article 276. <https://doi.org/10.1007/s10462-024-10925-w>
6. Hou, H., Jawaddi, S. N. B. A., & Ismail, A. (2024). Energy efficient task scheduling based on deep reinforcement learning in cloud environment: A specialized review. *Future Generation Computer Systems*, 151, 214–231. <https://doi.org/10.1016/j.future.2023.10.002>
7. Sunkara, S. P. (2025). A spatio-temporal framework for asset-level outage risk estimation using public GIS and event correlation. *International Journal of Computer Engineering and Technology*, 16(1), 4211–4227. https://doi.org/10.34218/IJCET_16_01_286
8. Kalai Arasan, K., & Anandhakumar, P. (2023). Energy-efficient task scheduling and resource management in a cloud environment using optimized hybrid technology. *Software: Practice and Experience*. <https://doi.org/10.1002/spe.3203>
9. Verma, P., Maurya, A. K., & Yadav, R. S. (2024). A survey on energy-efficient workflow scheduling algorithms in cloud computing. *Software: Practice and Experience*, 54(5), 637–682. <https://doi.org/10.1002/spe.3292>
10. Zhou, G., Tian, W., Buyya, R., Xue, R., & Song, L. (2024). Deep reinforcement learning-based methods for resource scheduling in cloud computing: A review and future directions. *Artificial Intelligence Review*, 57, Article 124. <https://doi.org/10.1007/s10462-024-10756-9>
11. Yu, X., Mi, J., Tang, L., Long, L., & Qin, X. (2025). Dynamic multi objective task scheduling in cloud computing using reinforcement learning for energy and cost optimization. *Scientific Reports*, 15, Article 45387. <https://doi.org/10.1038/s41598-025-29280-z>
12. S. K. Ramareddy, "Integrating AI-Driven Data Visualization and Data Warehouse Analytics for Predictive Decision-Making in Financial and Stock Market Systems," 2026 International Symposium of Systems, Advanced Technologies and Knowledge (ISSATK), Hammamet, Tunisia, 2026, pp. 1-6, doi: 10.1109/ISSATK69667.2026.11566934.