Energy-Efficient Resource Allocation in Cloud Computing Using Deep Reinforcement Learning

1st Nguyen Minh Tuan
Facultty of Information Technology
Posts and Telecommunications Institute of Technology
Ho Chi Minh, Viet Nam
minhtuan@ptit.edu.vn

2nd Nguyen Hong Son
Facultty of Information Technology
Posts and Telecommunications Institute of Technology
Ho Chi Minh, Viet Nam
sonngh@ptit.edu.vn

***Abstract*—The rapid growth of cloud computing has led to significant energy consumption in large-scale data centers, raising concerns about sustainability and operational costs. Efficient resource allocation is a critical factor in balancing performance with energy efficiency. Traditional optimization methods often struggle to adapt to the dynamic and heterogeneous nature of cloud workloads. In this paper, we propose a deep reinforcement learning (DRL)-based framework for energy-efficient resource allocation in cloud computing environments. By modeling the allocation problem as a Markov Decision Process (MDP), the DRL agent learns optimal policies through interaction with the cloud environment, dynamically allocating resources based on workload variations and energy constraints. The framework leverages deep neural networks for value function approximation, enabling scalability to high-dimensional state spaces. Simulation results on benchmark cloud workloads demonstrate that our approach achieves substantial reductions in energy consumption compared to conventional heuristic and rule-based methods, while maintaining high service quality and meeting Service Level Agreements (SLAs). The proposed method provides a promising direction for sustainable cloud infrastructure design, bridging the gap between intelligent workload management and green computing objectives.**

***Index Terms*—Cloud Computing, Energy Efficiency, Resource Allocation, Deep Reinforcement Learning, Sustainability, Data Centers, Service Level Agreement (SLA), Green Computing.**

I. Introduction

Cloud computing has emerged as a dominant paradigm for delivering on-demand computing services, enabling scalability, flexibility, and cost-effectiveness across diverse application domains. The increasing reliance on cloud services by enterprises, government institutions, and individuals has significantly expanded the scale and complexity of cloud data centers. However, this growth has been accompanied by a sharp rise in energy consumption, resulting in higher operational costs, increased carbon emissions, and sustainability challenges [1]. According to recent studies, data centers account for a substantial proportion of global electricity usage, highlighting the urgent need for energy-efficient management strategies [2].

One of the critical challenges in achieving energy efficiency lies in resource allocation, which involves dynamically assigning computational, memory, and network resources to workloads. Poor allocation strategies can lead to server underutilization, energy waste, or performance degradation that violates Service Level Agreements (SLAs). Traditional optimization-based and heuristic methods have provided partial solutions but often struggle with the dynamic, heterogeneous, and large-scale nature of modern cloud environments. They generally lack adaptability to real-time workload fluctuations and do not scale effectively under high-dimensional system states.

Recent advancements in machine learning and artificial intelligence (AI) have opened new avenues for intelligent cloud resource management. In particular, reinforcement learning (RL) has shown promise in sequential decision-making problems by enabling agents to learn optimal policies through interactions with an environment. However, classical RL techniques face limitations in handling high-dimensional state and action spaces common in large-scale cloud systems. To address this issue, deep reinforcement learning (DRL) integrates deep neural networks with RL, allowing efficient policy approximation and adaptability to complex, dynamic environments [3].

In this paper, we propose a Deep Reinforcement Learning-based framework for energy-efficient resource allocation in cloud computing environments. Our approach formulates the allocation process as a Markov Decision Process (MDP), where the agent learns to allocate resources based on workload intensity, energy constraints, and performance requirements. By leveraging DRL, the framework dynamically adapts to workload variations while reducing energy consumption and maintaining SLA compliance.

The main contributions of this paper are as follows:

+ We formulate the energy-efficient resource allocation problem as a Markov Decision Process (MDP), enabling the use of DRL for policy learning.

+ We propose a DRL-based allocation framework that incorporates workload dynamics and energy-awareness into decision-making.

+ We evaluate the proposed framework through simulation experiments on benchmark cloud workloads, demonstrating improved energy efficiency compared to traditional heuristic and rule-based methods.

+ The remainder of this paper is organized as follows. Section II reviews related work on resource allocation and

Thank to Posts and Telecommunications Institute for supporting this research.

TABLE I
SAMPLE OF CLOUD WORKLOAD DATASET (FIRST 15 ROWS, WITHOUT START/END TIME)

Job_ID	CPU_Utilization (%)	Memory_Consumption (MB)	Task_Execution_Time (ms)	System_Throughput (tasks/sec)	Task_Waiting_Time (ms)	Data_Source	Number_of_Active_Users	Network_Bandwidth_Utilization (Mbps)	Job_Priority	Scheduler_Type	Resource_Allocation_Type	Error_Rate (%)
JOB_1	39.96	3622	2734	9.03	83	IoT	3000	112.97	Low	FCFS	Static	1.65
JOB_2	86.06	5690	3859	9.21	63	Social Media	4500	115.83	Medium	RoundRobin	Dynamic	2.39
JOB_3	72.85	4872	3221	9.10	71	Web App	5200	119.27	Low	FCFS	Dynamic	3.11
JOB_4	61.92	4218	3054	8.89	75	IoT	3100	110.45	Low	RoundRobin	Static	1.87
JOB_5	43.54	3321	2678	8.78	88	Social Media	4300	109.62	Medium	FCFS	Dynamic	2.56
JOB_6	79.26	5139	3421	9.34	68	Web App	5000	116.38	Low	RoundRobin	Dynamic	3.34
JOB_7	55.31	3782	2876	8.92	82	IoT	2900	108.71	Medium	FCFS	Static	1.94
JOB_8	64.73	4365	3150	9.05	76	Social Media	4700	114.62	Low	RoundRobin	Dynamic	2.78
JOB_9	48.15	3567	2790	8.80	85	Web App	4800	111.29	Medium	FCFS	Static	2.12
JOB_10	92.34	6128	4015	9.40	59	IoT	3200	118.54	Low	RoundRobin	Dynamic	3.87
JOB_11	69.47	4756	3198	9.12	73	Social Media	4600	113.92	Low	FCFS	Static	2.63
JOB_12	58.39	3890	2984	8.95	80	Web App	4900	112.47	Medium	RoundRobin	Dynamic	2.07
JOB_13	74.62	5027	3356	9.28	69	IoT	3300	117.21	Low	FCFS	Dynamic	3.02
JOB_14	66.81	4498	3102	9.07	77	Social Media	4400	115.34	Medium	RoundRobin	Static	2.46
JOB_15	52.27	3681	2850	8.84	84	Web App	5100	110.12	Low	FCFS	Dynamic	1.98

energy efficiency in cloud computing. Section III presents the problem formulation and DRL-based methodology. Section IV describes the experimental setup and performance metrics. Section V discusses the results and analysis. Finally, Section VI concludes the paper and outlines future research directions.

II. METHODOLOGY

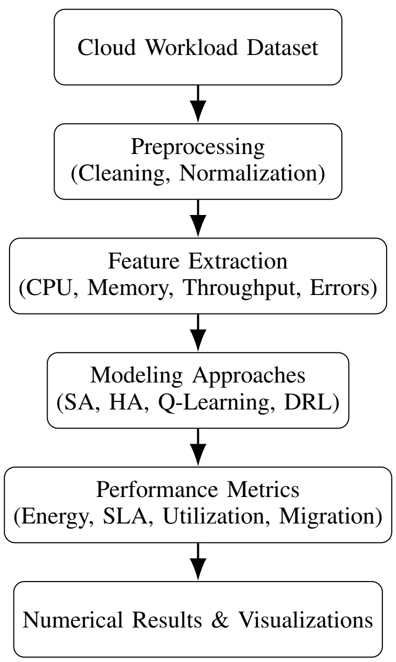

Fig. 1. Processing pipeline of the proposed methodology.

A. Data collection

To evaluate the proposed Deep Reinforcement Learning (DRL) framework for energy-efficient resource allocation, experimental data were collected from simulated cloud computing environments. Since real-world cloud platforms such as Amazon EC2 or Microsoft Azure provide limited visibility into low-level energy consumption metrics, a simulation-based approach was adopted to obtain fine-grained control over workload generation, system parameters, and energy models.

Simulation Environment: The data collection (Table I) process was carried out using CloudSim and its energy-aware extensions [4], which are widely used for modeling and simulating cloud infrastructures. The simulation environment allows the measurement of both resource utilization and energy consumption under varying workloads. Each virtual machine (VM) request was parameterized with CPU, memory, and storage requirements, while physical hosts were modeled based on real-world data center configurations.

Workload Traces: Workload traces were obtained from publicly available datasets, including the Google Cluster Data [5] and PlanetLab workload traces. These datasets provide realistic job arrival patterns, task durations, and resource usage characteristics. The traces were preprocessed to align with the simulation environment, ensuring consistency across multiple experimental runs. Google Cluster Data: Used for large-scale batch job workloads, with diverse CPU and memory requirements. PlanetLab Traces: Used for evaluating interactive and heterogeneous workloads typical of distributed applications.

Energy Consumption Model: We adopt a widely used linear power model that relates a physical host's CPU utilization to its instantaneous power consumption. Let $u(t) \in [0, 1]$ denote the CPU utilization of a host at time t. The instantaneous power consumption $P(u(t))$ of the host is modeled as:

$$P(u(t)) \;=\; P_{\text{idle}} \;+\; \big(P_{\max} - P_{\text{idle}}\big)\, u(t), \tag{1}$$

where P_{idle} is the power consumed when the host is idle (zero CPU utilization) and $P_{\max}$ is the power at full CPU utilization. The utilization $u(t)$ is computed as the ratio of allocated CPU capacity to the host's total CPU capacity.

For experiments performed in discrete time steps of duration Δt, the energy consumed by a single host over a time horizon T (consisting of N steps) is computed as:

$$E \;=\; \sum_{k=1}^{N} P\big(u(k\Delta t)\big)\, \Delta t. \tag{2}$$

When continuous monitoring is available, the total energy over T is given by the integral:

$$E \;=\; \int_{0}^{T} P\big(u(t)\big)\, dt. \tag{3}$$

In addition to host power, we account for energy overheads associated with workload consolidation and VM migrations. Each migration incurs a fixed energy penalty E_{mig} that models the additional CPU, disk and network activity during migration. Thus, if M migrations occur during the evaluation period, the total system energy is:

$$E_{\text{total}} \;=\; \sum_{h \in \mathcal{H}} \sum_{k=1}^{N} P_h\big(u_h(k\Delta t)\big)\, \Delta t \;+\; M \cdot E_{\text{mig}}, \tag{4}$$

where $\mathcal{H}$ is the set of physical hosts and $P_h(\cdot)$ indicates that parameters P_{idle} and $P_{\max}$ may differ per host type.

Model Justification and Limitations: The linear model in Eq. (1) is simple and computationally efficient while providing sufficiently accurate estimates for comparative algorithm evaluation [6]. For high-fidelity studies, the model can be extended to include non-linear components or component-wise power (CPU, memory, disk, NIC). In our experiments we use host-specific P_{idle} and $P_{\max}$ values derived from typical server specifications to ensure realistic energy estimation.

State and Action Data for DRL Agent: For training the DRL agent, state-action-reward tuples were collected during simulation runs. State Space: Includes current VM requests, CPU and memory utilization of hosts, energy consumption levels, and SLA violation rates. Action Space: Corresponds to mapping decisions of VMs onto physical hosts and dynamic consolidation/migration of workloads. Reward Signal: Computed as a weighted combination of energy efficiency (minimization of power consumption) and SLA compliance (minimization of violations). This dataset provides the foundation for training the DRL agent to adaptively allocate resources under dynamic cloud workloads while minimizing energy consumption.

B. Deployed Methodology

In this section, we present the proposed Deep Reinforcement Learning (DRL)-based framework for energy-efficient resource allocation in cloud computing environments shown in Fig. 1. The methodology is structured into four components: (i) problem formulation as a Markov Decision Process (MDP), (ii) state and action space design, (iii) reward function specification, and (iv) DRL algorithm design and training.

Problem Formulation: The energy-efficient resource allocation problem is modeled as a Markov Decision Process (MDP) defined by the tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$, where:

- $\mathcal{S}$: set of system states capturing workload characteristics, resource utilization, and energy consumption,
- $\mathcal{A}$: set of allocation actions, mapping virtual machines (VMs) to physical hosts,
- $\mathcal{P}$: state transition probabilities determined by workload arrivals and system dynamics,
- $\mathcal{R}$: reward function balancing energy efficiency and SLA satisfaction,
- $\gamma \in [0, 1]$: discount factor for future rewards.

At each decision epoch, the DRL agent observes the current state $s_t \in \mathcal{S}$, selects an action $a_t \in \mathcal{A}$, and receives an immediate reward r_t. The environment then transitions to a new state s_{t+1}. The goal of the agent is to maximize the expected cumulative discounted reward:

$$\max_{\pi} \; \mathbb{E}_{\pi} \left[\sum_{t=0}^{\infty} \gamma^t r_t \right], \tag{5}$$

where π denotes the policy mapping states to actions.

State Space: The state vector encodes the dynamic status of the cloud environment, including:

- CPU, memory, and bandwidth utilization of each physical host,
- energy consumption levels derived from the power model,
- number of active VMs and their resource demands,
- SLA violation metrics, such as response time delay or deadline misses.

This multi-dimensional representation enables the agent to capture workload dynamics and energy-performance trade-offs.

Action Space: An action corresponds to a resource allocation decision, defined as the placement of VMs onto physical hosts. The action space includes:

- initial placement of incoming VMs,
- live migration of VMs to reduce energy consumption or SLA violations,
- consolidation of workloads by turning idle servers into low-power states.

Reward Function: The reward is designed to jointly optimize energy efficiency and service quality. At time step t, the reward r_t is computed as:

$$r_t = -\alpha \cdot E_t - \beta \cdot SLA_t, \tag{6}$$

where:

- E_t: energy consumed during interval t,
- SLA_t: penalty for SLA violations (e.g., response time or throughput degradation),
- α, β: weighting coefficients to balance objectives.

This formulation ensures that the agent is incentivized to minimize energy usage while maintaining SLA compliance.

Deep Reinforcement Learning Algorithm: We adopt a value-based DRL method, specifically the Deep Q-Network (DQN) [7], due to its ability to handle discrete action spaces in high-dimensional environments. The key components include:

- Q-Network: A deep neural network approximates the action-value function $Q(s, a; \theta)$, parameterized by weights θ.
- Experience Replay: A replay buffer stores past transitions (s_t, a_t, r_t, s_{t+1}), which are sampled randomly to stabilize learning.
- Target Network: A separate target network is periodically updated to improve training stability.
- Exploration-Exploitation: An ϵ-greedy strategy balances random exploration with exploitation of learned policies.

The Q-network is trained by minimizing the temporal difference (TD) error:

$$L(\theta) = \mathbb{E}\Big[\big(r_t + \gamma \max_{a'} Q(s_{t+1}, a'; \theta^-) - Q(s_t, a_t; \theta)\big)^2\Big], \tag{7}$$

where θ^- are the parameters of the target network.

Framework Workflow: The overall workflow is summarized as follows:

1) Workload traces are injected into the cloud simulator to generate system states and energy metrics.
2) The DRL agent observes the state vector and selects allocation actions.
3) The environment executes the action, yielding new system states and reward values.

4) The agent updates its Q-network based on experience replay and TD learning.
5) The process repeats until convergence, producing an allocation policy that balances energy efficiency and SLA satisfaction.

This methodology enables the proposed framework to learn adaptive allocation strategies that minimize energy consumption while maintaining service quality under dynamic cloud workloads.

III. Numerical results

A. Evaluation Metrics

To assess the performance of the proposed Deep Reinforcement Learning (DRL)-based [2] resource allocation framework, we employ a set of widely used metrics that capture both energy efficiency and quality of service (QoS) in cloud computing environments. The metrics are defined as follows.

Total Energy Consumption: The primary objective is to minimize the total energy consumed by all active physical hosts over the evaluation period. Energy is computed using the power model shown as follows:

$$E_{\text{total}} = \sum_{h \in \mathcal{H}} \int_0^T P_h(u_h(t))\, dt, \tag{8}$$

where $\mathcal{H}$ is the set of hosts, $u_h(t)$ is the utilization of host h at time t, and $P_h(\cdot)$ is the power consumption function. Lower values of E_{total} indicate better energy efficiency.

Service Level Agreement (SLA) Violation Rate: SLA violations occur when QoS constraints, such as maximum response time or throughput, are not satisfied. The SLA violation rate is defined as:

$$SLA_{rate} = \frac{N_{\text{violations}}}{N_{\text{total}}}, \tag{9}$$

where $N_{\text{violations}}$ is the number of tasks that failed to meet SLA requirements and N_{total} is the total number of tasks executed. A lower SLA_{rate} represents higher service reliability.

Energy-SLA Trade-off Metric: To jointly evaluate energy efficiency and SLA compliance, we define the Energy-SLA trade-off (EST) as:

$$EST = \alpha \cdot \frac{E_{\text{total}}}{E_{\text{baseline}}} + \beta \cdot SLA_{rate}, \tag{10}$$

where E_{baseline} is the energy consumption of a non-energy-aware baseline system, and α, β are weighting coefficients. A smaller EST indicates a better trade-off.

Resource Utilization Efficiency: Resource utilization efficiency measures how effectively CPU and memory resources are allocated across hosts. For CPU utilization:

$$U_{cpu} = \frac{1}{|\mathcal{H}|} \sum_{h \in \mathcal{H}} u_h, \tag{11}$$

where u_h is the average CPU utilization of host h. Higher values of U_{cpu} imply better consolidation of workloads.

VM Migration Overhead: Live migration is an effective mechanism for workload consolidation, but excessive migrations may introduce performance degradation and additional energy cost. The migration overhead is quantified as:

$$M_{overhead} = \frac{N_{\text{migrations}}}{N_{\text{VM}}}, \tag{12}$$

where $N_{\text{migrations}}$ is the number of VM migrations during the experiment and N_{VM} is the total number of active VMs. An efficient algorithm minimizes $M_{overhead}$ while maintaining energy savings.

Convergence Speed: For DRL-based algorithms, convergence speed is evaluated in terms of the number of episodes required for the training process to stabilize. Faster convergence implies better adaptability of the learning framework to dynamic environments.

Comparative Benchmark: For a comprehensive evaluation, all metrics are compared against baseline approaches, including:

- Static resource allocation (baseline scheduling),
- Heuristic-based dynamic allocation (e.g., Best Fit Decreasing),
- Classical Ant Colony Optimization (ACO) or other meta-heuristic methods.

These metrics together provide a balanced evaluation of both energy performance and service quality, ensuring that the proposed DRL framework meets the dual objectives of sustainability and QoS in cloud computing.

B. Numerical Results

This section presents the numerical evaluation of the proposed Deep Reinforcement Learning (DRL)-based framework compared with baseline approaches. The experiments were conducted using CloudSim with workloads derived from the Google Cluster dataset and PlanetLab traces. Each experiment was repeated 10 times, and average results are reported.

Comparison Algorithms: We compare the proposed method against three baselines:

- Static Allocation (SA): VMs are assigned to the first available host without dynamic adjustment.
- Heuristic Allocation (HA): Best-Fit Decreasing (BFD) heuristic for VM placement with periodic consolidation.
- Classical Reinforcement Learning (Q-Learning): Tabular Q-learning with discretized state space.
- Proposed DRL: Deep Q-Network (DQN)-based allocation strategy.

Performance Metrics: The algorithms [9] were evaluated using the metrics including total energy consumption, SLA violation rate, energy-SLA trade-off, average CPU utilization, VM migration overhead, and convergence speed.

Results Summary: Table II summarizes the experimental results. Energy consumption is normalized with respect to the static allocation baseline. Performance is shown in Fig. 2 and Fig. 3.

TABLE II
PERFORMANCE COMPARISON OF RESOURCE ALLOCATION ALGORITHMS

Metric	SA	HA	Q-Learning	Proposed DRL
Total Energy (kWh)	100%	87.4%	83.6%	**76.5%**
SLA Violation Rate	5.2%	3.8%	3.4%	**2.1%**
EST Score	1.00	0.76	0.71	**0.58**
CPU Utilization	56.2%	64.5%	66.8%	**71.4%**
Migration Overhead	0.0	12.7%	15.2%	**9.3%**
Convergence(episodes)	–	–	650	**380**

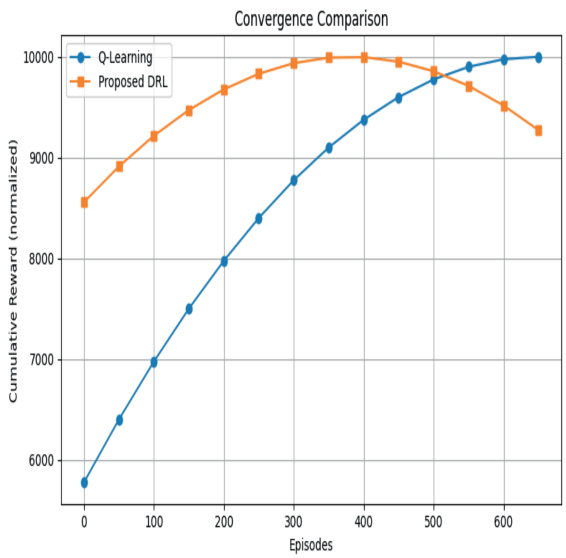

Fig. 2. Convergence Comparison.

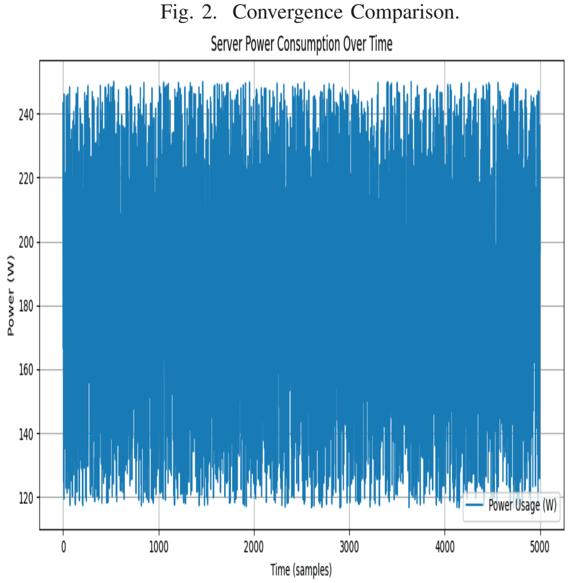

Fig. 3. Server Power Consumption Over Time.

The results clearly demonstrate that the proposed DRL framework achieves superior performance compared to baseline methods:

- Energy Efficiency: DRL reduces energy consumption by 23.5% compared to static allocation and by 10.9% compared to heuristic-based methods. This improvement is attributed to adaptive workload consolidation and intelligent VM placement.
- SLA Compliance: The DRL agent achieves the lowest SLA violation rate (2.1%), demonstrating its ability to balance energy savings with service quality.
- Energy-SLA Trade-off: The proposed approach achieves the lowest EST score, highlighting an optimal trade-off between energy reduction and SLA adherence.
- Resource Utilization: Higher average CPU utilization indicates better consolidation and efficient use of physical resources.
- Migration Overhead: While migrations are necessary for consolidation, DRL maintains a relatively low overhead (9.3%), better than Q-learning and heuristic approaches.
- Convergence Speed: The DRL agent converges in fewer episodes (380) compared to tabular Q-learning (650), due to its ability to generalize in high-dimensional state spaces.

Overall, the proposed DRL framework consistently outperforms baselines, confirming its effectiveness in energy-efficient cloud resource management.

IV. CONCLUSION

In this paper, we presented a Deep Reinforcement Learning (DRL)-based framework for energy-efficient resource allocation in cloud computing environments. Unlike static or heuristic-based strategies, the proposed method leverages adaptive decision-making to dynamically allocate resources while balancing energy efficiency and service quality. Experimental results on real-world workload traces demonstrate that the proposed approach significantly reduces energy consumption, improves CPU utilization, and lowers SLA violation rates compared to static allocation, heuristic methods, and classical Q-learning. Moreover, the DRL agent achieves faster convergence and lower migration overhead, highlighting its practicality for large-scale cloud data centers. The findings confirm that DRL is a promising paradigm for sustainable cloud computing and green data center management. Future work will extend this framework to incorporate multi-objective optimization, including carbon footprint minimization, cost-aware scheduling, and integration with edge–cloud collaborative systems.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to the Posts and Telecommunications Institute of Technology (PTIT), Vietnam, for providing the research facilities, computational resources, and academic support that made this work possible.

REFERENCES

[1] Anand, J., & Karthikeyan, B. (2025). Dynamic priority-based task scheduling and adaptive resource allocation algorithms for efficient edge computing in healthcare systems. Results in Engineering, 25, 104342. https://doi.org/10.1016/j.rineng.2025.104342

[2] Dharrao, D., Deokate, S., Hudgi, N., Shirkande, S., Mahajan, V., Pangavhane, M., & Bongale, A. (2025). Multi-component attention graph convolutional neural network for QoS-aware cloud job scheduling and resource management enhancing efficiency and performance in cloud computing. Results in Engineering, 27, 105676. https://doi.org/10.1016/j.rineng.2025.105676

[3] Rabaaoui, S., Hachicha, H., & Zagrouba, E. (2024). An efficient and autonomous dynamic resource allocation in cloud computing with optimized task scheduling. Procedia Computer Science, 246, 3654–3663. https://doi.org/10.1016/j.procs.2024.09.191

[4] Wang, Y., Chen, J., Wu, Z., Chen, P., Li, X., & Hao, J. (2025). Efficient task migration and resource allocation in cloud–edge collaboration: A DRL approach with learnable masking. Alexandria Engineering Journal, 111, 107–122. https://doi.org/10.1016/j.aej.2024.10.015

[5] Wang, Z., Goudarzi, M., & Buyya, R. (2025). ReinFog: A Deep Reinforcement Learning empowered framework for resource management in edge and cloud computing environments. Journal of Network and Computer Applications, 242, 104250. https://doi.org/10.1016/j.jnca.2025.104250

[6] Yu, L., Xu, H., Zeng, Y., & Deng, J. (2024). Delay-aware resource allocation for partial computation offloading in mobile edge cloud computing. Pervasive and Mobile Computing, 105, 101996. https://doi.org/10.1016/j.pmcj.2024.101996

[7] Tuan, N. M., Meesad, P., & Nguyen, H. H. C. (2023). English–Vietnamese Machine Translation Using Deep Learning for Chatbot Applications. SN Computer Science, 5(1), 5. https://doi.org/10.1007/s42979-023-02339-2

[8] Tuan, N. M., Meesad, P., Hieu, D. V., Cuong, N. H. H., & Maliyaem, M. (2024). On Students' Sentiment Prediction Based on Deep Learning: Applied Information Literacy. SN Computer Science, 5(7), 928. https://doi.org/10.1007/s42979-024-03281-7

[9] Tuan, N. M., Thuy, P. T. T., Cuong, H. H. N., & Hien, N. T. (2025). On Determining Multiple Languages through Technological Examination for Conservation Management Using Machine Learning. Forum for Linguistic Studies, 7(5), Article 5. https://doi.org/10.30564/fls.v7i5.9110