Dynamic Cloud Resource Optimization Using Reinforcement Learning

[1]Prof. S. A. Bachwani., [2]Dr. D. N. Chaudhari., [3]Shreya R. Gothe., [3]Aditi A. Trivedi., [3]Ojas Kharate

[1]Research Scholar, Department of Computer Science and Engineering, Sant Gadge Baba Amravati University, JDIET Yavatmal, India

[2]Professor & Dean, Head of Department of Computer Science and Engineering, Jawaharlal Darda Institute of Engineering and Technology Yavatmal, India

[3]Department of Computer Science and Engineering, Jawaharlal Darda Institute of Engineering and Technology Yavatmal, India

DOI: https://doi.org/10.51244/IJRSI.2026.1309000321

Received: 07 October 2026; Accepted: 12 October 2026; Published: 28 October 2026

ABSTRACT

Cloud computing offers flexible and scalable access to computational resources, but selecting a suitable resource configuration for a given workload remains a difficult task. High-Performance Computing (HPC) and Artificial Intelligence (AI) workloads have varying computational requirements, and their performance depends on complex interactions between workload characteristics and the available cloud configurations. Traditional resource recommendation methods generally rely on predefined rules, repeated test executions, or large volumes of historical data, which are poorly suited to new or changing applications. This paper presents a study of dynamic cloud resource optimization based on reinforcement learning, using the Oikonomos-II+ framework as its primary technical reference. In this approach, resource selection is treated as a contextual multi-armed bandit problem in which workload characteristics form the context and the available cloud configurations form the possible actions. A Variational Autoencoder (VAE) produces a compact representation of workload characteristics, Neural-LinUCB selects configurations while balancing exploration and exploitation, and Gaussian Process (GP) regression estimates expected performance together with its uncertainty. RHO-LOSS supports the evaluation of resource-selection outcomes. After a configuration is selected, the workload is executed, its performance is observed, and the resulting feedback is used to update the learning models and improve future recommendations. The paper reviews traditional, machine-learning-based and reinforcement-learning-based allocation approaches, explains the methodology and working cycle of the framework, and discusses its applications, advantages, challenges, limitations and future scope. Overall, the study concludes that learning-based, workload-aware recommendation is a promising route to better resource utilization and suitable performance for computationally intensive workloads, although computational overhead, data quality, implementation complexity and scalability require further evaluation.

Keywords: Cloud Resource Optimization, Reinforcement Learning, Contextual Bandit, Neural-LinUCB, Gaussian Process

INTRODUCTION

Cloud computing has become an important paradigm for delivering computational resources such as processing capacity, memory, storage, networking and specialized accelerators over a network. A major characteristic of the cloud is elasticity: the amount of allocated resources can be increased or decreased as workload requirements change. This flexibility, however, creates a management challenge. A cloud environment may offer a very large number of resource configurations, and a configuration that performs well for one application may be inefficient for another. Over-allocation leaves resources underutilized and increases cost, whereas under-allocation degrades performance or prevents an application from meeting its requirements [3], [10].

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue IX September 2026

Traditional resource management relies on predefined rules, thresholds and manually designed policies. These techniques work well for predictable workloads but adapt poorly when workload behavior changes frequently [2], [3]. Machine learning can identify patterns in workload and performance data, yet resource allocation is not purely a prediction problem: the system must choose one action from several alternatives and then observe the consequence of that choice. Reinforcement learning (RL) addresses this setting directly, because an agent learns which actions suit which situations through repeated interaction with its environment [5]. RL has been investigated for resource allocation and scheduling in cloud-edge systems, distributed machine learning and wireless networks [1], [6], [8]–[10], and the wider literature on resource management and on scaling RL has been surveyed in [3] and [7].

This paper studies dynamic cloud resource optimization using reinforcement learning, with the Oikonomos-II+ recommender [2] as its main technical reference. Oikonomos-II+ combines contextual multi-armed bandits, Neural-LinUCB and Gaussian Process regression to recommend cloud resources for HPC and AI workloads, and also considers workload characterization and feedback-based resource evaluation.

Problem statement

Selecting an inappropriate resource configuration can result in underutilization, unnecessary cost or degraded performance, and static allocation approaches may not adapt to changing workload characteristics [2], [3]. There is therefore a need for an approach that understands workload characteristics, considers multiple candidate configurations, learns from earlier allocation decisions, balances the exploration of new configurations with the exploitation of configurations that already perform well, and uses feedback to improve future decisions.

Objectives and scope

The objectives of this study are to examine the challenges of dynamic cloud workloads; to explain how reinforcement learning and contextual decision-making can be applied to resource recommendation; to describe the roles of Neural-LinUCB, Gaussian Process regression, the Variational Autoencoder and RHO-LOSS; to analyze the exploration–exploitation balance; and to discuss the applications, advantages, limitations and future possibilities of the approach. The paper is a study of the methodology and architecture presented in the reference work; it does not claim to implement a complete production resource-management platform.

Traditional cloud resource allocation

Traditional approaches depend on predefined policies, thresholds and heuristics; for example, additional CPU is allocated when utilization exceeds a set level. They are simple to implement and effective when workload behavior is predictable. However, a fixed threshold cannot suit every application, and a rule-based system has limited ability to learn from earlier decisions: if a configuration performs poorly for a workload, it may continue to be selected until the policy is changed manually [2], [3].

Machine learning-based resource management

Machine learning uses information such as CPU and memory utilization, execution time and application behavior to support workload classification, resource-demand prediction, performance prediction, anomaly detection, capacity planning and cost optimization. A prediction-based system can estimate the resources a workload requires before execution. Resource optimization, however, involves more than prediction, because the system must select one configuration from several alternatives and determine whether that choice produced a desirable outcome. This motivates approaches that combine prediction with decision-making [2], [6].

Reinforcement learning-based approaches

Reinforcement learning provides a framework in which an agent learns by observing the results of its actions rather than from labelled examples; Q-learning, one of its best-known families of algorithms, is classified in [5]. In cloud resource management the agent selects a configuration for a workload, observes the resulting performance and uses it as feedback. The process can be described as observe, decide, allocate, execute, measure and learn. Deep RL has been applied to resource allocation and scheduling in cloud-edge collaboration [6],

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue IX September 2026

distributed machine learning [9] and edge-cloud multi-resource allocation [10], and RL-based resource optimization has also been studied for heterogeneous networks [4], RAN slicing [1] and massive MIMO scheduling [8]. The central difficulty is deciding how much to explore unknown configurations and how much to rely on those already known to perform well.

Contextual multi-armed bandits

A contextual multi-armed bandit selects an action based on the current context and learns from the resulting reward [2]. In cloud resource optimization the available configurations are the arms and the characteristics of the current workload are the context. For every workload one configuration is selected and its outcome observed, so the system gradually learns which configurations suit which contexts. This is important because, for example, a CPU-intensive workload may benefit from greater CPU capacity whereas a memory-intensive one may benefit from larger memory [2].

Comparison of existing approaches

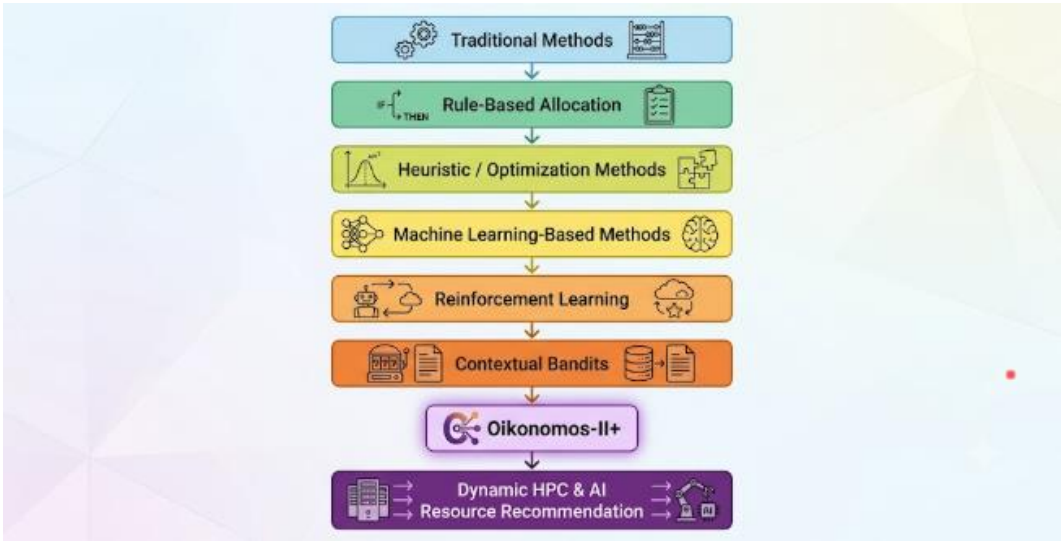
Table 1 compares the main families of approaches. Traditional and heuristic methods are simple but less adaptive, while machine-learning and RL-based methods offer greater flexibility for complex problems [3], [7], [9]. Oikonomos-II+ [2] extends this direction by combining a modified Neural-LinUCB algorithm with Gaussian Process regression for online instance recommendation for HPC and AI workloads. Unlike approaches that depend on historical training data or repeated auxiliary executions, it is designed to learn online from submitted jobs [2]. The evolution of these approaches is summarized in Figure 1.

Table 1: Comparison of cloud resource allocation approaches

Approach	Main technique	Adaptability	Major limitation
Traditional allocation	Static rules / predefined policies	Low	Limited response to changing workloads
Heuristic and meta-heuristic methods	Heuristics / optimization algorithms	Moderate	May require repeated optimization and tuning
Predictive ML-based methods	Machine learning / prediction models	Moderate	Depends on quality and availability of training data
RL-based autoscaling	Reinforcement learning	High	Training and exploration overhead
Deep RL-based scheduling	Deep reinforcement learning	High	Computational and implementation complexity
Oikonomos-II+	Neural-LinUCB + Gaussian Process regression	High	Requires continuous learning and model management
Studied approac	RL-based dynamic resource optimization	High	Scalability and prediction accuracy need further evaluation

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue IX September 2026

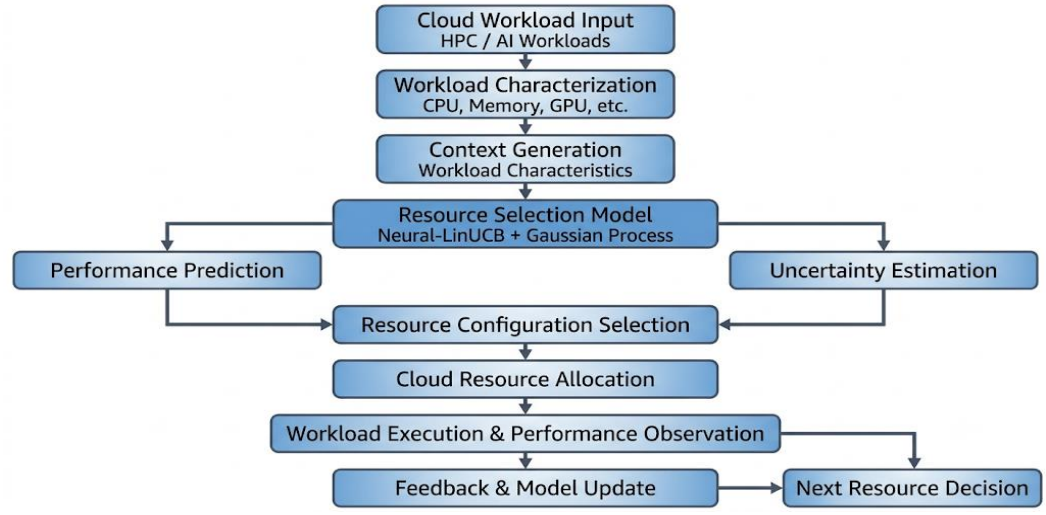
Figure 1: Evolution of cloud resource allocation approaches



METHODOLOGY

The methodology is based on the Oikonomos-II+ framework [2], which uses reinforcement-learning-based resource recommendation for HPC and AI workloads. Instead of fixed allocation rules, the system uses workload information and earlier resource–performance observations to choose a configuration that balances performance and resource usage [2]. The overall pipeline is shown in Figure 2: workload, workload characterization, resource selection, execution, performance feedback and learning.

Figure 2: Methodology of dynamic cloud resource optimization



Workload characterization

Different workloads have different computational requirements, so the first step is to characterize the workload. A Variational Autoencoder (VAE), a generative neural-network model, learns a compact, lower-dimensional representation of the workload data (workload data, encoder, latent representation). This representation is used as contextual information by the recommendation mechanism, which allows the system to distinguish between workload types instead of treating every workload as identical [2].

Contextual resource selection

Resource selection is formulated as a contextual multi-armed bandit problem. The workload representation is the context and the available resource configurations are the arms; Table 2 summarizes the elements. This permits workload-specific decisions instead of applying one configuration to all workloads [2].

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue IX September 2026

Table 2: Elements of the contextual bandit formulation

Element	Description
Context	Characteristics of the workload
Arms	Available resource configurations
Action	Selected resource configuration
Feedback	Observed workload performance

Neural-LinUCB and Gaussian process regression

Neural-LinUCB combines neural-network-based representation learning with an upper-confidence-bound strategy. It balances exploitation, which selects configurations already known to perform well, with exploration, which tries configurations about which little is known. Gaussian Process (GP) regression complements it by estimating the expected performance of a configuration together with the uncertainty of that estimate, so decisions consider both the predicted value and how reliable it is [2].

Feedback, RHO-LOSS and model update

After a configuration is selected, the workload is executed with the allocated resources and its performance is observed. This observation is the feedback that updates the learning models. RHO-LOSS is incorporated in the selection process, and its loss-based evaluation helps assess the quality of resource-selection decisions. Through repeated interaction, the relationship between workload characteristics and resource configurations is learned and future recommendations improve [2].

Overall procedure

The complete procedure can be summarized in the following steps:

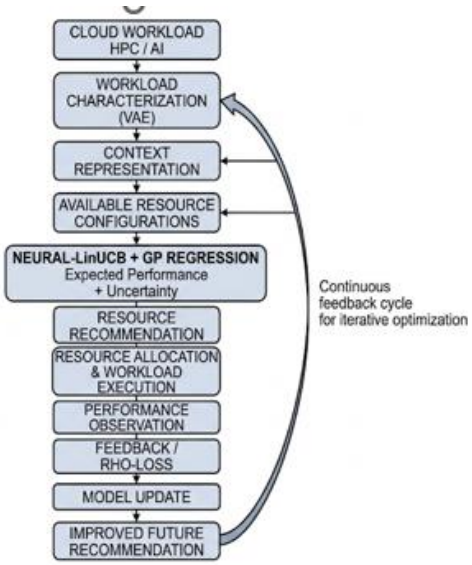
1. A workload is received by the cloud resource-management system.
2. Its characteristics are extracted and represented using the VAE.
3. The representation is used as the context for resource selection.
4. Available configurations are evaluated; Neural-LinUCB selects one while balancing exploration and exploitation, and GP regression estimates performance and uncertainty.
5. The selected resources are allocated and the workload is executed.
6. The resulting performance is observed and used as feedback.
7. The learning models are updated and the updated knowledge is used for future recommendations.

Working of the framework

The working of Oikonomos-II+ is an adaptive cycle, shown in Figure 3. A submitted HPC or AI workload is characterized by the VAE, and the resulting context, together with the available configurations, is passed to the Neural-LinUCB and GP regression component, which considers expected performance and uncertainty. The recommended configuration is allocated, the workload runs, and performance is observed. If the outcome is desirable, the system gains evidence supporting that configuration for similar contexts; otherwise the observation helps improve later decisions. Because the cycle repeats for every workload, the recommender continuously adapts to changing workload behavior [2].

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue IX September 2026

Figure 3: Working process of Oikonomos-II+



DISCUSSION

Exploration and exploitation

The exploration–exploitation trade-off is central to the approach. A system that only exploits known configurations may never discover a better one, while excessive exploration repeatedly selects poor or uncertain configurations. If Configuration A has performed well for a workload while Configuration B has been tested only a few times, the system will usually prefer A but may occasionally try B because it could perform better. By incorporating uncertainty into the decision, the contextual bandit formulation with Neural-LinUCB handles this situation in a principled way [2], [5].

Applications

The approach is applicable wherever workloads have varying computational requirements: HPC workloads, AI and machine-learning workloads, cloud-based scientific computing, data-intensive applications and dynamic cloud infrastructure shared by multiple workloads. Its primary application is the development of cloud-resource recommendation systems that help users or administrators select suitable configurations, reducing dependence on manual selection and fixed policies, and supporting decisions that consider both cost and performance [2], [3], [10].

Advantages

- **Workload-aware allocation:** recommendations follow the characteristics of the current workload, which is more flexible than fixed policies and helps avoid unnecessary allocation [2].
- **Adaptive learning:** feedback from earlier decisions improves later ones, and accumulated experience supports faster decisions once sufficient learning has taken place [2], [9].
- **Controlled exploration:** insufficiently evaluated configurations can be explored without relying only on previously selected resources [2], [5].
- **Uncertainty-aware prediction:** GP regression supplies predicted performance and uncertainty, which is useful when historical information is limited [2].
- **Reduced manual effort:** selection patterns are learned from data rather than from hand-tuned rules and thresholds [2], [10].

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue IX September 2026

- **Suitability for complex workloads:** the approach suits HPC and AI workloads and heterogeneous resources, and supports continuous optimization [2], [3].

Challenges and limitations

- **Dynamic workloads and large decision space:** workloads change over time, and the number of possible configurations makes the selection problem complex [3], [10].
- **Limited initial information:** early recommendations may be less accurate until enough observations are collected, and exploration may temporarily yield inefficient allocations [2], [5].
- **Data quality and monitoring:** incomplete or inconsistent data weakens learning, and continuous monitoring and storage of workload and performance information add overhead [3], [6].
- **Computational overhead and complexity:** processing workload information, updating models and integrating several learning components is more demanding than rule-based allocation, and parameter choices affect prediction quality [7], [9].
- **Scalability and adaptation:** larger numbers of workloads and configurations may require additional optimization, and models must adapt as workloads and resources change [3], [7].
- **Prediction uncertainty:** even with GP regression, predictions can be inaccurate when few observations exist for a workload–resource combination [2].

Future scope

Future research can improve the adaptability, scalability and practical deployment of intelligent resource-management systems in the following directions:

- Real-time monitoring of workloads and infrastructure so that recommendations respond faster to changing conditions [3].
- Integration with cloud orchestration platforms to allocate, scale and release resources automatically [3], [10].
- Support for a wider range of heterogeneous resources, including CPUs, GPUs, memory, storage and network resources [3], [4].
- Extension to multi-cloud and hybrid-cloud environments by comparing configurations across providers [3].
- Joint optimization of performance, resource utilization and cloud cost [4].
- Investigation of more advanced RL models and continuous, automated learning for larger and more dynamic problems [1], [6], [8].
- Improved scalability of the recommendation process for very large numbers of workloads, users and configurations [7].

CONCLUSION

Dynamic cloud resource optimization using reinforcement learning offers an effective approach to improving resource allocation, particularly for complex HPC and AI workloads. By combining workload characterization, contextual bandit techniques, Neural-LinUCB and Gaussian Process regression, the Oikonomos-II+ approach recommends configurations while considering both expected performance and uncertainty, and it adapts to changing conditions through feedback from earlier decisions [2]. This can contribute to better resource

utilization, workload-aware decision making and continuous optimization of cloud infrastructure. The approach also involves challenges, namely computational overhead, dependence on quality data, implementation complexity, exploration risk, scalability and the need for continuous model adaptation [2], [7]. Further improvements in scalability, accuracy and real-world deployment will make such approaches more practical for future cloud computing environments.

ACKNOWLEDGEMENTS

The authors thank the Department of Computer Science and Engineering and the management of Jawaharlal Darda Institute of Engineering & Technology, Yavatmal, for providing the facilities needed to carry out this work.

Conflict of interest

The authors declare that they have no known competing financial interests or personal relationships that could have influenced the work reported in this paper.

Data availability

No new datasets were generated or analyzed in this study. The paper is based on previously published literature, which is listed in the references.

Ethical approval

Not applicable. This study did not involve human participants or animals.

REFERENCES

1. Akyıldız, H. A., Gemici, Ö. F., Hökelek, I., & Çırpan, H. A. (2024). Hierarchical reinforcement learning based resource allocation for RAN slicing. IEEE Access, 12, 75818–75831.
2. Betting, R., Chen, Q., De Zeeuw, C., & Strydis, C. (2026). Oikonomos-II+: A reinforcement-learning, cloud-resource recommender for HPC & AI workloads. IEEE Transactions on Cloud Computing, 14(2), 1141–1157.
3. Boubaker, N. E. H., Zarour, K., Guermouche, N., & Benmerzoug, D. (2025). A comprehensive survey on resource management for IoT applications in edge-fog-cloud environments. IEEE Access, 13, 111892–111925.
4. Gnanasekar, S., & Sriharipriya, K. C. (2026). Environmentally aware quantum-inspired reinforcement learning framework for green power and resource optimization in 6G heterogeneous networks. IEEE Transactions on Green Communications and Networking, 10, 3473–3484.
5. Jang, B., Kim, M., Harerimana, G., & Kim, J. W. (2019). Q-learning algorithms: A comprehensive classification and applications. IEEE Access, 7, 133653–133667.
6. Li, Q., et al. (2024). Deep reinforcement learning based resource allocation for fault detection with cloud edge collaboration in smart grid. CSEE Journal of Power and Energy Systems, 10(3), 1220–1230.
7. Liu, D., et al. (2024). Scaling up multi-agent reinforcement learning: An extensive survey on scalability issues. IEEE Access, 12, 94610–94631.
8. Lopes, V. H. L., et al. (2022). Deep reinforcement learning-based scheduling for multiband massive MIMO. IEEE Access, 10, 125509–125525.
9. Lu, X., et al. (2023). RLPTO: A reinforcement learning-based performance-time optimized task and resource scheduling mechanism for distributed machine learning. IEEE Transactions on Parallel and Distributed Systems, 34(12), 3266–3280.
10. Qadeer, A., & Lee, M. J. (2023). Deep-deterministic policy gradient based multi-resource allocation in edge-cloud system: A distributed approach. IEEE Access, 11, 20381–20398.