Vol. 08, No. 2 (2026) 1113-1122, doi: 1010.24874/PES08.02B.002

Proceedings on Engineering Sciences

www.pesjournal.net

AN INTELLIGENT REINFORCEMENT LEARNING-DRIVEN FRAMEWORK FOR PREDICTIVE RESOURCE TRACKING AND ALLOCATION IN CLOUD COMPUTING ENVIRONMENTS

Bakkala Santha Kumar[1]
R Shankar

Received 11.02.2026.
Revised 25.05.2026.
Accepted 05.06.2026.

Keywords:

Cloud computing,
Energy-aware scheduling,
Hardware resource tracking,
Predictive data analytics,
Resource allocation,
Reinforcement learning

A B S T R A C T

Cloud computing has become a formidable paradigm of computing services that are on-demand and with scale. Nevertheless, efficient resource monitoring and allocation is an element that poses a very critical issue especially when the hardware parts that are monitored and allocated are heterogeneous and partially opaque (obscure) and the components are spread widely in extensive cloud systems. Poor visibility and poor distribution of such resources usually result to a fall in performance, high cost of operation and inefficiency in energy use. To solve these problems, the present paper suggests an intelligent framework consisting of automated hardware monitoring at the hypervisor level and predictive resource provisioning with the use of reinforcement learning. The suggested solution achieved through experimental approach is useful in providing the ability to properly manage the tracing of the elusive hardware devices even when there is a parallel event triggering and dynamic workload scenario.

© 2026 Published by Faculty of Engineering

1. INTRODUCTION

Cloud computing has been an essential technology forming the basis of the contemporary digital ecosystems and scaling on demand, access to computing, storage and networking. Business organizations are progressively basing their operations on cloud-based platforms to pursue data intensive applications, time sensitive analytics, and business critical services. Despite these strengths, the skyrocketing nature of cloud environments has attained abundant challenges pertaining to resource visibility, dynamically allocated resources, and effective usage of the resources especially in large-scale and heterogeneous environments.

Modern cloud systems have an enormous amount of hardware and virtual resources whose characteristics of capacity, performance, and operating conditions differ. Numerous of these resources, particularly those functioning at the hypervisor or infrastructure level, have limited levels of visibility and dynamicity and consequently can be hard to monitor and control through traditional monitoring systems. Lack of effective monitoring of these little-known hardware elements tend to result in overlapping provisioning, contention of resources, sluggish execution of services and high operational expenses. Under parallel event triggered and varying workloads, these problems are compounded as

[1] Corresponding author: Bakkala Santha Kumar
Email: santhakumar.bakkala@gmail.com

the allocated systems cannot respond well using rule-based or fixed allocation policies. To prevent these shortcomings, intelligent self-adaptive resource management structures that integrate real-time monitoring, predictive, and learning-based decision-making mechanisms are becoming increasingly necessary. Machine learning and specifically reinforcement learning (RL) have come in as an encouraging solution to simulating the complex cloud environments and learning the best allocation policies by constantly engaging and interacting with the system. Reinforcement learning combined with predictive analytics using historic workload trends can predict resource needs ahead of time and continuously revise, depending on current conditions, the strategy to allocate resources and keep the performance and Quality of Service (QoS) high.

Here, the current work suggests a smart learning-based reinforcement learning predictive resource tracking and share in the cloud computing environment. The given research model combines automated hypervisor-level monitoring of tracking obscure hardware resources and predictive analytics of demand forecasting and adaptive allocation, as an example, reinforcement learning. An intermediary component, which is broker based, coordinates requests of resources, tracks system conditions and implements learned allocation policies. The architecture will maximize scalability, decrease energy use, decrease idle resources, and meet the requirements of QoS with a dynamic workload environment.

The following hypotheses will be used to guide the study based on this research model:

- Predictive allocation which is based on reinforcement learning can greatly enhance the overall use of the resources by improving the utilization when compared to the greedy strategy and the reactive one.
- Regeneration of redundant provisioning and overhead caused by complicated hardware resources is minimized by automated monitoring.
- The energy-conscious policies under reinforcement learning are associated with quantifiable energy savings and cost savings with quality-of-service guarantees.
- The framework suggested improves the robustness and adaptability of the system in the event of abrupt changes in the workload and the failure of the resources.

2. LITERATURE REVIEW

The past ten years saw cloud resource management shift its focus towards non-intelligent, rule-driven provisioning to the more intelligent (learning-based) orchestration models. In the 2020-2025 research, the researchers put an increasing focus on the application of machine learning (ML) and deep reinforcement learning (DRL) to the shortcomings of the traditional heuristics, especially in large-scale, multi-tenant, and highly dynamic cloud environments. Dominant trends in surveys that have recently been done illustrate that current cloud infrastructures need adaptive decision-making engines that could maximize several competing goals, such as performance, cost, energy usage, and Quality of Service (QoS) compliance (Zhou et al., 2024; Khan et al., 2022).

The early works of this era were mainly devoted to the workload prediction and capacity planning based on classical models of ML and deep learning. Recurrent neural network-based and attention mechanism-based time-series forecasting methods showed a greater level of demand prediction capability relative to the reactive threshold-based ones (Khan et al., 2022). These predictive techniques, however, were found to be inadequate with very bursty workloads and non-portable usage patterns and do not provide feedback-based adaptation procedures. This led scientists to implement prediction models alongside closed-loop control approaches to achieve better robustness and responsiveness (Chen et al., 2022). The reinforcement learning has become a strong paradigm in allocating cloud resources because it has the capacity to learn the best policy by continually interacting with the environment. It has been shown that Q-learning, Deep Q-Networks and actor-critic all are effective in task scheduling, virtual machine placement and auto-scaling (Zhou et al., 2024; Rossi et al., 2023). These strategies allow dynamically changing strategies of resource allocation in response to observed performance and performance indicators to prevent SLA violations and enhance the overall utilization. However, most of the current solutions based on RL assume fully observable environments and do not consider the issues of heterogeneous and partially opaque hardware resources. In recent years, energy efficiency and cost optimization have been one of the most popular research topics, as the operational cost and environmental impact of big data centers have been increasing. To reduce the energy consumption, but at the same time acceptable execution time, deep reinforcement learning-based task scheduling and VM consolidation methods have been suggested (Hou et al., 2024; Shaw et al., 2022). Despite these developments, most investigations consider resource tracking and allocation as two different activities, which have restricted the efficiency of the policies that are learned in case resource state details are stale or incomplete.

Related to the emergence of cloud-native underpinnings, microservices, containers, and serverless computing, resource management has now become more complicated due to the new complexities introduced. According to the recent studies, the conventional CPU-centric scaling plans do not adequately apply to such

environment with the performance of the applications being depending on more than just a few metrics as request rate, tail latency, and inter-service dependencies (Zhang et al., 2021; Baarzi & Kesidis, 2021). Auto-scaling mechanisms based on machine learning and reinforcement learning have been put forward to overcome these hurdles and have been shown to be more adaptable than their counterparts, which are not auto-scaling (Pintye et al., 2024; Taha et al., 2024). Nonetheless, these solutions tend to need fine-grained observability and precise monitoring of underlining hardware and virtual resources to operate successfully. The other significant line of research is the edge-cloud and IoT-integrated environments where resource allocation models must be able to consider latency, mobility, and energy constraints on distributed infrastructures. Recent literature suggests multi-objective and federated learning models of reinforcement to schedule tasks and allocate resources in edge-cloud systems, with great potential in terms of scalability and privacy-preservation (Zhang & Ou, 2025). However, these solutions tighten even more the necessity of automated and intelligent monitoring of heterogeneous resources at the various levels of the infrastructure.

Even though it has made significant progress, the literature continues to highlight various outstanding challenges. These are: training instability and convergence problems in DRL models, lack of explainability on the learned policies, lack of sufficient integration between predictive analytics and reinforcement learning, and insufficient mechanisms of tracking obscure or low-visibility hardware resources (Zhou et al., 2024, Rossi et al., 2023). According to recent publications, the management systems of cloud resources must take the form of unified, closed-loop, systems, with real-time monitoring, predictive modelling and reinforcement learning to deliver reliable, energy-efficient, and cost-effective operation (Chen et al., 2022; Quattrocchi et al., 2024; Toka et al., 2021; Dogani et al., 2023). The current studies show the prospects of learning-based resources allocation and only a big gap in integrated structures that synergistically tackle resource tracking, predictive demand estimates, and learning-based allocation through adaptive reinforcement learning. This gap has been bridged in the current work which proposes an intelligent, energy aware framework that integrates these components to provide scalability, robustness, and efficiency to current cloud computing environment.

3. RESEARCH METHODOLOGY

In this section, the experimental setting, properties of the workload, system architecture, and baseline approaches, as well as measurement of evaluation, to substantiate the proposed Adaptive Energy-Aware Prediction and Reinforcement (AEPR) prototyping to monitor and allocate cloud resources are detailed.

3.1 Experimental Setup

Experiments were all carried out within a cloud environment that was virtualized in Python. It had to use a simulator to allow controlled, repeatable and scalable experimentation without the need to access a production cloud infrastructure. The method is popular in the studies on cloud resource management, especially the assessment of scheduling and allocation policies in various workload scenarios. The models of the simulated cloud infrastructure are a homogeneous pool of virtual machines (VMs) with capacity dimensions that are fixed and homogeneous. The decisions on the allocation of resources are made at predetermined time intervals during which the system monitors the demand of workloads, resources, and tracking data after which they decide on scaling measures. This environment is connected to the reinforcement learning agent in the form of a typical Markov Decision Process (MDP). The experimental model combines four closely coupled modules, including workload creation and consumption, resource tracking in a partially observable manner, demand prediction and resource allocation through reinforcement learning.

3.2 Workload Modelling

To test the strength of the suggested framework in realistic working circumstances, a synthetic workload generator created the demand of the cloud applications. The workload generated captures timer-averaged CPU demand as well as captures the important properties of real cloud traces such as periodic behavior, slow demand changes, stochastic noise, and bursts. The workload includes several thousand steps in discrete time, each having a decision point of resource allocation. Periodic components represent the lifetime operation of diurnal and short-term use, whereas random burst injections represent flash-crowds and the sudden bursts of traffic. The design will give such a workload an opportunity to demonstrate both non-stationary and stationary properties of the workload which is going to enable measuring the predictive and adaptive abilities of the suggested system. This is because the same workload trace is employed in all the proposed and baseline methods to compare them fairly.

3.3 Resource Tracking and Partial Observability

To model the difficulty of tracing vague or low-visibility hardware resources, an extra resource tracing strata was implemented in between the workload source and the allocation controller. This layer models the imperfection of realistic monitoring that usually occurs in a large-scale cloud system. Particularly, the tracker presents three categories of uncertainty, including incomplete visibility, when telemetry is unavailable every couple of seconds, delayed observations, when received metrics are not equal to the actual resources; and noisy measurements,

when seen values are not equal to real demand because of instrumentation errors. The tracker contains an estimated demand at any point as well as visibility flag which shows whether telemetry is available at a certain point at a particular time step. These tracking artifacts are expressly defined as part of the state of the reinforcement learning agent, and it is possible to learn allocation policies that are imperfectly observable.

3.4 Demand Prediction Model

Lightweight sliding-window workload demand prediction model is used to predict the demand of short-term workload. The predictor believes in the latest historical developments to infer the demand at the forthcoming decision-making position. Linear regression is selected as the minimum forecasting model with the reason that it is simple, easily understandable and least demanding to compute. Periodically, a predictor is retrained as a part of the simulation process to exhibit changes in the workload patterns. The system state can be defined in terms of the predicted values of demand that are provided to the agent of the reinforcement learning, which allows implementing not only reactive but also proactive decisions regarding resources allocation. The individual performance of the predictions is measured by conventional measures of regressions errors to guarantee that the accuracy of the forecasts is high enough to aid in downstream allocation choices.

3.5 Reinforcement Learning Environment and Agent Configuration

The problem of the cloud resource allocation is demonstrated as a Markov Decision Process. In every step, the agent takes observations of the system state now and chooses a discrete action, among the three available options, namely, scaling down resources, keeping the existing allocation, or scaling up resources. The state variable consists of the demand recorded, demand predicted, normalized VM number, tracking visibility reflector, the current level of utilization and normalized latency reflector. The reward role is developed to promote compliance to SLA and effective use of resources, and discourage SLA non-compliance, high idle resource use, and hard scale behavior. The reinforcement learning agent is also trained by the Proximal Policy Optimization (PPO) algorithm because it is stable, efficient in its samples, and it generally performs well in continuous control and resource management challenges. The offline training on the simulated environment is done, followed by the assessment of the trained policy in inference mode.

3.6 Proposed AEPR Framework, Evaluation Metrics and Evaluation Procedure

To prove the effectiveness of the suggested AEPR framework, the comparison with a number of available baseline allocation strategies that are widely used in cloud systems is provided a threshold policy which manages resources upon observed utilized thresholds, a prediction-only policy that acts based on the demand forecasts and not learning to reinforcement, a random allocation policy, in which scaling actions are chosen in a random manner, and the policy proposed that is a policy of reinforcement learning that incorporates tracking and prediction. The methods are all subjected to the same workloads and configurations of the systems. The effectiveness of each approach is measured with the help of various measures comprising Quality of Services, efficiency, cost, and stability. These are SLA violation rate, average and tail latency, average resource utilization, total operational cost proxy, energy consumption proxy and scaling churn. Such an overall measure would ensure that gains in one dimension will not be achieved at the cost of others.

4. EXPERIMENTS AND RESULTS

All the experiments are conducted in one standardized procedure. The workload trace is first initialized, and the predictor is then trained on a sample of historical data. The allocation policy is then implemented, and the metrics of the system are observed at every decision point in time of a specified number of time steps. The experiment is repeated on the baselines, and the offered method and integrated metrics are estimated to compare them. Table 1 provides complete experimental setup including parameters and other configurations.

A detailed analysis of the suggested Adaptive Energy-Aware Prediction and Reinforcement (AEPR) system based on the simulation environment is discussed. The suggested approach is contrasted with three base line policies namely a threshold-based autoscaler, a prediction-only controller, and a random scaling policy. The performance is evaluated regarding compliance to Quality of Service (QoS), latency behaviour, resource utilization, operational cost, energy efficiency and scaling. The methods are all compared on the same workload trace and decision horizon to provide a chance of fair comparison.

Table 1. Experimental parameters and configurations

Category	Parameter	Value / Description
Workload	Number of time steps	3,000–3,500
	Base demand level	0.45 (normalized)
	Seasonal amplitude	0.20
	Burst probability	0.02
	Burst magnitude	0.35
	Noise standard deviation	0.03
Tracking Layer	Telemetry visibility probability	0.88
	Observation delay	4 time steps
	Measurement noise	0.04
Cloud Infrastructure	VM capacity	0.12 demand units
	Minimum VMs	1
	Maximum VMs	60

QoS Model	Base latency	70 ms
	Latency growth slope	320 ms
	SLA threshold	200 ms
Cost & Energy	VM cost per step	0.002 (proxy units)
	Idle resource penalty	0.001
	Energy coefficient	0.04
Prediction Model	Model type	Sliding-window Linear Regression
	Window size	12 time steps
RL Agent	Algorithm	Proximal Policy Optimization (PPO)
	Learning rate	3×10^{-4}
	Discount factor (γ)	0.99
	GAE λ	0.95
	Training steps	60,000
Decision Making	Action space	{Scale down, Hold, Scale up}
	Decision interval	1 time step
Evaluation	Test horizon	1,500 steps
	Metrics	SLA violations, latency, utilization, cost, energy

4.1 QoS Compliance and Latency Analysis

The compliance with QoS can be measured based on the SLA violation rate that is the percentage of deciding steps when the realized latency was above SLA threshold. As it can be seen in Table 2, threshold-based and prediction-only policies have zero SLA violations in the considered workload, which implies conservative behavior about workload provisioning. This is however at the cost of efficiency as discussed later. The random policy has a much larger SLA violation rate at 4.93 per cent which validates this hypothesis that uninformed scaling decisions cannot work well with workload variability. In comparison, the suggested model of AEPR has a very minimal violation of SLA of 0.2, which means that even in the case of unperfect resource tracking and slow telemetry, reinforcement learning can ensure the violation of QoS near-perfect compliance. This observation is also backed by the latency statistics. The mean latency of the proposed RL policy =72.81 ms, and this value is not too large as compared to the baseline latency floor but, still, works to reduce the extreme latency spikes. The 95 th -percentile latency of 86.50 ms is much lower than the random policy latency (173.79 ms), and thus, tail latency behavior is better.

Table 2. QoS and Latency Performance comparison

Method	**SLA Violation (%) ↓**	**Avg Latency (ms) ↓**	**P95 Latency (ms) ↓**
Threshold-based	**0.00**	**70.16**	**70.00**
Prediction-only	**0.00**	**70.00**	**70.00**
Random	**4.93**	**151.42**	**173.79**
Proposed RL (PPO)	**0.20**	**72.81**	**86.50**

Figure 1 Time lag about all the assessed policies. The suggested RL policy considerably reduces the latency spikes and shows constant performance relative to the random baseline. Figure 2 shows the correlation of the actual workload requirement and the estimated demand in the initial 400 test steps. The forecast model is a close reflection of the workload dynamics, and it managed to capture periodic trends and gradual demand changes. They are also characterized by minor deviations in the case of abrupt burst events, which is to be expected of a lightweight sliding-window regression model. However, this prediction precision is decent enough to give meaningful proactive cues to the allocator that relies on reinforcement learning to minimize the time taken to react to increased or decreased workloads.

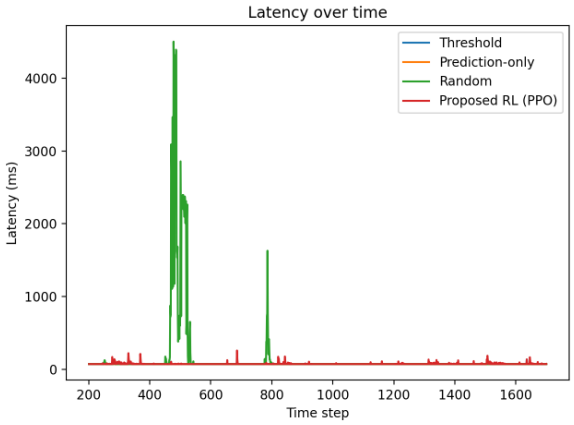


Figure 1. Latency over time for all evaluated policies

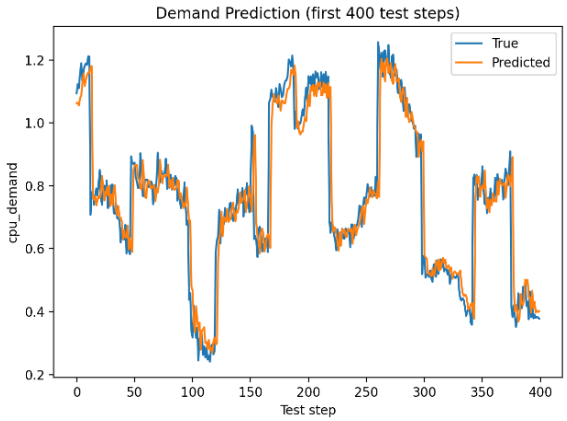


Figure 2. Comparison of true and predicted workload (demand prediction for the first 400 test steps)

4.2 Resource Utilization and Provisioning Efficiency

Although compliance with QoS is necessary, it is also equally vital to have an efficient use of resources to operate a cloud sustainably. According to Table 3, prediction-only policy would experience an incredibly low average utilization (10.24) although it will provide a very large number of VMs (56.93 VMs on average). The result of such behavior is a clear indication of excessive over-provisioning that is caused by modeling demand on

a conservative basis and no adaptive feedback control. The threshold-based policy will further enhance throughput to 54.37 percent but utilizes an average of 10.5 VMs which is reactive scaling behavior that does not utilize capacity fully. The random policy has a medium level of utilization (42.67) but is not stable and predictable. By contrast, the proposed AEPR framework has the best average utilization of 78.99% and the minimal average number of active VMs (6.83). This finding demonstrates the capacity of the learning reinforcement agent to trade off proactive scaling and effective consolidation by using both observed feedback and predicted demand even in the situation of partial observability.

Table 3. Resource Utilization and Provisioning Efficiency

Method	**Avg Utilization (%) ↑**	**Avg VMs ↓**
Threshold-based	54.37	10.50
Prediction-only	10.24	56.93
Random	42.67	15.94
Proposed RL (PPO)	78.99	6.83

Figure 3 shows the offered RL solution does not lead to QoS breaches as it sustains a higher utilization factor. Figure 4 shows the suggested RL policy is very sensitive to changes in the workload but does not rely on prolonged over-provisioning.

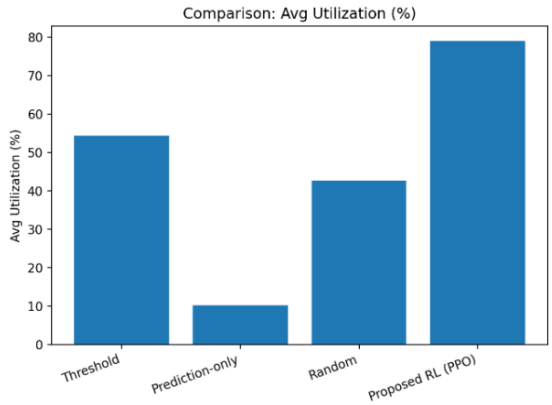


Figure 3. Trends in policy utilization

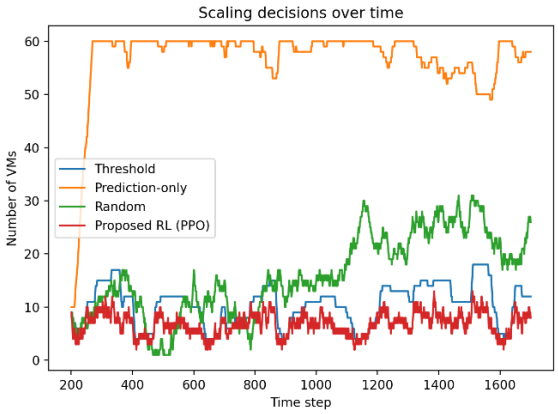


Figure 4. Provided VMs as a function of time

Figure 5 shows the trend in utilization with time of all the evaluated resource allocation policies. Platforms of prediction-only approach have an exemplary low utilization that is marked by aggressive over-provisioning whereas the random policy reveals uncertain and erratic utilization trends.

The policy of threshold attains moderate utilization even though it responds to workload changes in a conservative manner. Conversely, the policy of the proposed reinforcement learning ensures a constantly high utilization over time, which proves in its possibility to match the resource provisioning to the workload demand and exclude both overutilization and underutilization.

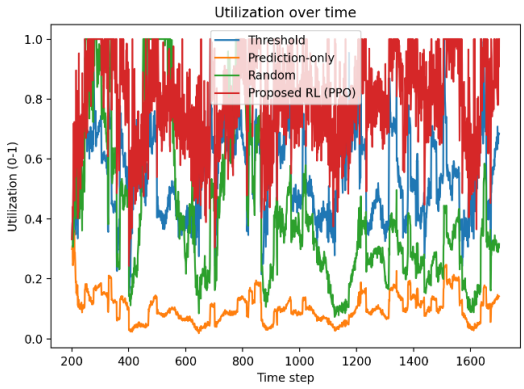


Figure 5. Efficient utilizations of resource allocation over time

4.3 Cost and Energy Efficiency

Operation cost and energy consumption are compared with proxy measures that capture VM-time utilization, idle resource allocation and utilization-weighted energy consumption. The prediction-only policy has the highest total cost (170.80) and energy usage (1416.79), as Table 4 indicates that it is associated with constant over-provisioning.

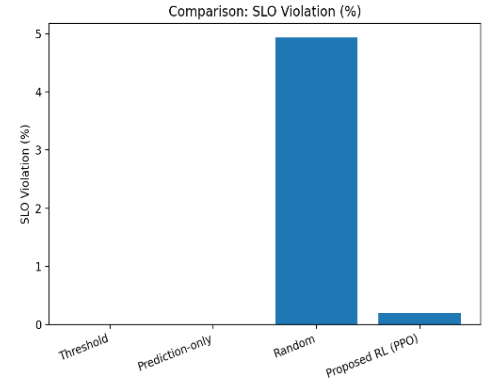
Table 4. Cost and energy consumption comparison

Method	**Total Cost (proxy) ↓**	**Total Energy (proxy) ↓**
Threshold-based	31.50	441.51
Prediction-only	170.80	1416.79
Random	47.82	546.77
Proposed RL (PPO)	**20.50**	**359.70**

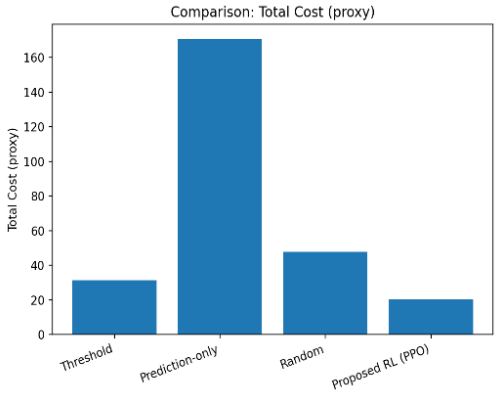
This points out the important weakness of prediction-based scaling without adaptive control. With random policy, the cost and the energy consumption is also high but this is mainly caused by unstable scaling behavior and inefficiency. The threshold-based policy provides average cost and energy performance with sub-optimal performance since it scales reactively. The proposed AEPR Non -A framework has the lowest overall cost (20.50), and the lowest overall proxy of energy (359.70) compared to all methods considered. These findings prove that combining reinforcement learning and

predictive signals with tracking awareness have significant economic and energy savings and can achieve almost perfect compliance with the QoS.

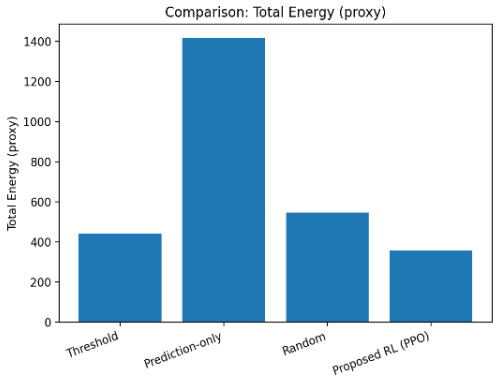
Figure 6 shows a comparative bar diagram of the SLA violations, utilization, proxy (cost), and energy proxy. The proposed RL model has always had a better trade-off in all the metrics.



(a)



(b)



(c)

Figure 6. Comparison of (a) SLA violations, utilization, (b) proxy (cost), and (c) energy proxy

5. DISCUSSIONS

One of the positive outcomes of this work is the clear inclusion of imperfect resource tracking. Tracking layer also adds jitter and noisy telemetry, as well as periodically unavailable ones, which simulates real-life problems of cloud monitoring. The classical baseline policy implicitly relies on trustworthy and real-time observations and as such fail to perform when the quality of telemetry becomes compromised. The visible indicator is built into the suggested AEPR framework and learns to adjust in scaling decisions depending on the confidence of observed data. This allows the agent to insure against uncertain observations, prevent overreaction to noisy signals, and remain stable in their performance. The findings show that a combination of learning-based allocation and tracking awareness and predictive cue is resistant to the partial observability and is better than the conventional choices in terms of all the dimensions considered.

- The proposed RL-based system has minimal SLA violations (0.2) with a minimal latency.
- It provides the most resource usage (78.99) and the limited number of active VMs (6.83).
- It costs the least of all threshold-driven scaling by about 35 percent and prediction-only scaling by the most by about 88 percent.
- It has the highest energy efficiency, which facilitates a green cloud operation.
- It also performs even in the case of imperfect tracking and delayed telemetry.

5.1 Future Scope

Although the suggested AEPR framework shows high performance, there are some directions that are yet to be pursued by future research. The first is a simulation assessment in the current study where homogeneous virtual machines are used to evaluate the framework. The framework is going to be expanded to heterogeneous cloud infrastructures in the future, where there are different types of VM and accelerators (such as GPUs) and multi-zone/ multi-region deployments. Second, the proxy models together with the models estimating cost and energy consumption make future research incorporate interfaces that measure hardware-level power or data center energy models to be able to optimize more specifically based on energy consumption. The reinforcement learning agent will be able to directly maximize actual energy expenditure as opposed to surrogate measures. Third, the reinforcement learning agent during the current implementation is centrally trained. Future studies will examine the concept of federated and distributed reinforcement learning methods to enable training and learning in large geographically distributed cloud and edge-cloud systems and maintaining data locality and privacy. Fourth, learning based cloud controllers are still facing problems of explainability as well as trust. Future extensions will also include explainable reinforcement learning methods to offer interpretable understanding of allocation choices and this will enhance operator trust and enable real-world implementation. Finally, the framework can be scaled to justify container based and serverless environments, where resource management that is finer-grained, high scaling and function-based scheduling, create even more

complexity. The combination of AEPR and Kubernetes schedulers, as well as serverless runtimes is a logical and effective next step in this work.

6. CONCLUSION

In this paper, an intelligent and single framework was introduced, as an Adaptive Energy-Aware Prediction and Reinforcement (AEPR), which is applicable in resource tracking and dynamic provision in the cloud computing environment. The offered solution is a solution to one of the most fundamental drawbacks of the traditional cloud resource management systems, which is the fact that they are based on the reactive or prediction-only approach wherein infrastructure resources are assumed to be perfectly observed. The AEPR framework allows extensive predictive analytics with reinforcement learning because it explicitly by modeling imperfect resource tracking and combines the predictive analytics analysis with reinforcement learning and under realistic cloud monitoring constraints. One of the advantages of the proposed framework is its holistic nature. Resource tracking at the infrastructure level is made automatic and reports the current state of resources even when there is sluggish or noisy telemetry. The short-horizon demand prediction provides proactive information that minimizes the response time to changes in workload, whereas the reinforcement learning agent updates allocation policies that maximize the goals of a long-term system. By balancing Quality of Service (QoS) compliance, resource use, cost of operation, and energy savings, the combination of such components enables the system to operate in a unit control cycle. There is a wide range of simulation-based experiments that prove the efficiency of the offered method. The AEPR structure shows almost zero levels of SLA violation and low mean and tails latency compared to threshold-based, prediction-only and random allocation policies. Meanwhile, it can contribute greatly to improving the use of resources and minimizing the number of working virtual machines, which will contribute positively to both the cost of work and the consumption of energy. Noteworthy, these advantages are achieved even with the partial observability, which proves the resilience of the learning-based policy when the resources are not tracked perfectly.

Competing Interests: The authors declare that there is no conflict of interest.

Funding Information: This research did not receive grants/funds from any public or commercial agencies.

Author contribution: Author1 conceptualized defining the research objectives by developing and implementing the integrated reinforcement framework. The author1 also conducted the process of data collection, conducting the data analysis, and conduct the experiments. Author2 initiated the manuscript writing and contributing to revisions and modifications for preparing the final manuscript. Author2 also revised the paper based on the research requirements and provided insights throughout the preparation of the manuscript.

Informed Consent: Not Applicable

References

Baarzi, A. F., & Kesidis, G. (2021). SHOWAR: Right-sizing and efficient scheduling of microservices. In *Proceedings of the 2021 ACM Symposium on Cloud Computing* (pp. 427–441). Association for Computing Machinery. https://doi.org/10.1145/3472883.3486999

Chen, X., Zhu, F., Chen, Z., Min, G., Zheng, X., & Rong, C. (2022). Resource allocation for cloud-based software services using prediction-enabled feedback control with reinforcement learning. *IEEE Transactions on Cloud Computing, 10*(2), 1117–1129. https://doi.org/10.1109/TCC.2020.2992537

Dogani, J., Namvar, R., & Khunjush, F. (2023). Auto-scaling techniques in container-based cloud and edge/fog computing: Taxonomy and survey. *Computer Communications, 209*, 120–150. https://doi.org/10.1016/j.comcom.2023.06.010

Hou, H., Jawaddi, S. N. 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

Khan, T., Tian, W., Zhou, G., Ilager, S., Gong, M., & Buyya, R. (2022). Machine learning (ML)-centric resource management in cloud computing: A review and future directions. *Journal of Network and Computer Applications, 204*, 103405. https://doi.org/10.1016/j.jnca.2022.103405

Pintye, I., Kovács, J., & Lovas, R. (2024). Enhancing machine learning-based autoscaling for cloud resource orchestration. *Journal of Grid Computing, 22*, Article 67. https://doi.org/10.1007/s10723-024-09783-1

Quattrocchi, G., Incerto, E., Pinciroli, R., & Trubiani, C. (2024). Autoscaling solutions for cloud applications under dynamic workloads. *IEEE Transactions on Services Computing, 17*(3), 804–820. https://doi.org/10.1109/TSC.2024.3354062

Rossi, F., Cardellini, V., Lo Presti, F., & Nardelli, M. (2023). Dynamic multi-metric thresholds for scaling applications using reinforcement learning. *IEEE Transactions on Cloud Computing, 11*(2), 1807–1821. https://doi.org/10.1109/TCC.2022.3163357

Shaw, R., Howley, E., & Barrett, E. (2022). Applying reinforcement learning towards automating resource allocation and application scalability in the cloud. *Information Systems, 107*, 101722. https://doi.org/10.1016/j.is.2021.101722

Taha, M. B., Sanjalawe, Y., Al-Daraiseh, A., Fraihat, S., & Al-E'mari, S. R. (2024). Proactive auto-scaling for service function chains in cloud computing based on deep learning. *IEEE Access, 12*, 38575–38593. https://doi.org/10.1109/ACCESS.2024.3375772

Toka, L., Dobreff, G., Fodor, B., & Sonkoly, B. (2021). Machine learning-based scaling management for Kubernetes edge clusters. *IEEE Transactions on Network and Service Management, 18*(1), 958–972. https://doi.org/10.1109/TNSM.2021.3052837

Zhang, W., & Ou, H. (2025). Reinforcement learning based multi objective task scheduling for energy efficient and cost effective cloud edge computing. *Scientific Reports, 15*, Article 25666. https://doi.org/10.1038/s41598-025-25666-1

Zhang, Y., Hua, W., Zhou, Z., Suh, G. E., & Delimitrou, C. (2021). Sinan: ML-based and QoS-aware resource management for cloud microservices. In *Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems* (pp. 167–181). Association for Computing Machinery. https://doi.org/10.1145/3445814.3446693

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

Bakkala Santha Kumar
Department of Computer Science and Engineering,
Koneru Lakshmaiah Education Foundation, Vaddeswaram,
Andhra Pradesh,
India
santhakumar.bakkala@gmail.com
ORCID 0009-0007-4739-0498

R Shankar
Department of Computer Science and Engineering,
Koneru Lakshmaiah Education Foundation,
Vaddeswaram,
Andhra Pradesh,
India
shankarrajendran75@kluniversity.in
ORCID 0000-0003-3509-4732