OPEN ACCESS
World Journal of Multidisciplinary Studies
ISSN (Online): 3049-0855 | Vol. 2, No. 10, Oct. 2025
Website: https://wasrpublication.com/index.php/wjms

AI-Driven Orchestration Framework for Cloud Computing Platforms

Rajalingam Malaiyalan*

Independent Researcher, USA.

Received: 20/05/2025 | Accepted: 07/09/2025 | Published: 21/10/2025

Abstract: The rise in the number of workloads in the cloud computing has been exponentially increasing and the growing complexity of multi-cloud implementations require the use of intelligent orchestration mechanisms that can no longer be achieved through standard rule-based automation. The AI-driven Orchestration Framework (ADOF) offered in this paper is a cloud computing platform that combines Long Short-Term Memory (LSTM) networks to predict the workload, Deep Q-Network (DQN) reinforcement learning to schedule the resources dynamically, and a federated privacy engine to train the cross-tenant models in a secure way. The framework proposed will consider a five-layer hierarchical structure which will include resource pooling, infrastructure abstraction, orchestration control, AI intelligence, and application interface layers. An experimental assessment of a multi-cloud testbed of heterogeneous clusters (which comprises heterogeneous resources and operational frameworks) proves that ADOF can lead to a 35.9 percent increase in the utilization of the resources, a 43.1 percent decrease in the average response time, and a 26.6 percent drop in the cost of operations as compared to traditional rule based orchestration baselines. The framework also demonstrates strong SLA compliance in the conditions of bursty workload and seamless integration with the existing frontend-back backend AI systems. These findings highlight the potential of the transformative nature of integrating learning-based intelligence into the cloud orchestration control planes.

Keywords: AI orchestration; cloud computing; deep reinforcement learning; LSTM workload prediction; multi-cloud management; federated learning; Kubernetes; resource scheduling.

1. Introduction

The current cloud computing environments have transformed simple virtualized server farms to multi-cloud environments that spread across geographically dispersed areas serving billions of simultaneous users. The result of this evolution has been new issues that are not typical in the field of resource management: workloads are changing at random, service-level agreements (SLAs) require response times of less than a second, and operational budgets must undergo constant cost optimization. Although traditional orchestration tools can automate discrete processes like provisioning of virtual machines and container deployment, they work by static policy, threshold-based policies, which cannot make effective use of the stochastic nature of newer cloud traffic patterns (Wang et al., 2025).

Artificial intelligence and cloud infrastructure have also been converged thus spawning a new breed of intelligent orchestration systems. In machine learning, especially deep learning and reinforcement learning versions, provide the ability to learn complex workload behavior based on historic data, forecast future demand behavior and dynamically adapt resource distribution in real time. Reinforcement learning-based auto scaling may dynamically adjust Kubernetes pod replicas based on changes in workload according to the requirements of the latter, as Mishra et al. (2024) showed that it outperforms reactive scaling policies by an order of magnitude in terms of throughput and latency.

Moreover, the introduction of AI into the cloud management has gone beyond mere predictive scaling. Modern systems will have to deal with the heterogeneous workload organization in multi-clouds, organize frontend-backend integration in business intelligence systems with AI-enhanced operations, and provide data privacy via federated learning (Talluri and Rachamala, 2023). In an effort to identify key architectural issues that the next-generation BI implementation should address, Talluri and Rachamala (2023) pointed out the need to orchestrate frontend analytics layers with AI inference engines at the back-end to ensure seamless operation of an analytics platform focused on AI.

In spite of these developments, there still exists an unified framework where a workload prediction, intelligent scheduling, anomaly detection, and privacy-conscious distributed learning all have been modeled into one orchestration control plane as an open research problem. Current solutions will focus on solving one part of cloud intelligence separately, and it is up to the system integrators to piece together unrelated tools with non-trivial compatibility overheads (Pintye et al., 2024).

In this paper, this gap is filled by proposing an AI-Driven Orchestration Framework to cloud computing platforms, called ADOF, that provides a comprehensive lifecycle of intelligent resources by providing a coherent and layered architecture. The main contributions of the work are:

***Corresponding Author**
Rajalingam Malaiyalan
Independent Researcher, USA.

1. A hierarchical and orchestration architecture with five layers that integrates resource pooling, infrastructure abstraction, AI intelligence, and control interfaces that are user friendly.

2. An LSTM-based workload prediction module that can predict CPU utilization with 92.4 percent accuracy horizons that span 30 minutes.

3. A reconfiguring resource scheduler that they use is a DQN-based resource scheduler, which optimizes task placement in heterogeneous cloud nodes dynamically.

4. An e-federated privacy engine that facilitates cooperative training of models among the cloud tenants without data centralization.

5. Strong improvements over rule-based baselines in utilization, latency and cost aspects on a heterogeneous testbed empirically tested and found to have significant results.

The rest of this paper will be structured in the following way. Section 2 is a literature review. Section 3 shows the framework architecture proposed. The experimental methodology is described in Section 4. Result and analysis are reported in section 5. Implications and limitations are talked about in section 6. The last section of section 7 contains future research directions.

2. Related Work

2.1 Machine Learning for Cloud Resource Management

The use of machine learning to manage cloud resources is a topic that has been researched during the last decade. Sarker (2021) gave a full taxonomy of different deep learning techniques that could be used in computing systems, which established vocabulary that would be used in future cloud-specific implementations. Initial investigations by researchers in references listed in Frontiers in Computer Science (2025) have shown that LSTM and related recurrent architectures are better on cloud workload time-series forecasting as compared to ARIMA based statistical predictors due to the ability to capture long-range temporal correlations without any explicit seasonality assumptions.

The design decisions made in the current work were directly inspired by a hybrid LSTM-DQN algorithm to allocate cloud resources that Wang et al. (2025) suggested and demonstrated an improvement of resource utilization by 32.5 percent and reduction of the response time by 43.3 percent in comparison with rule-based baselines. Mahimalur (2025) also showed that the hybrid reinforcement learning using deep neuronal networks enhanced the use of resources by 1822 percent and reduced latency by 15 percent in heterogeneous cloud-edge systems, indicating the scalability of learning-based methods.

The Pintye et al. (2024) designed an auto-scaling framework of cloud resources orchestration that adopted statistical feature selection to detect application-specific scaling metrics, and found that the different workload classes necessitate varying predictor settings. Their study on Journal of Grid Computing strengthened the role of metric diversity in the orchestration intelligence.

2.2 Deep Reinforcement Scheduling Learning

The ability to make decisions sequentially with uncertainty has seen deep reinforcement learning (DRL) become the leading paradigm of dynamic cloud scheduling. In their review, Springer Nature (2024) has enumerated DRL methods in the virtual machine migration, task offloading, and container scheduling domains, with the Deep Q-Network and Proximal Policy Optimization being the most popular algorithms. The review has observed that in the case of high stochasticity and multi-objective optimization needs, the DRL approaches always perform better than the heuristic and metaheuristic approaches.

Proposed by Bitsakos et al. (2025), DInos is an autoscaler of stateless cloud applications designed to consider the concept of deep reinforcement learning, which can receive 17.3x greater rewards in simulated tasks and a 5.5x greater increase in real Kubernetes deployments by using temporal workload modeling and pre-trained policies. Their work highlighted the importance of generalizable policies that experience minimum retraining in all contexts of deployment.

Mishra et al. (2024) confirmed the RL-based autoscaling implementation on an IEEE Cloud conference production environment and found that dynamic pod scaling policies trained using traffic traces of the Azure environment outperformed static threshold-based Kubernetes horizontal pod autoscalers (HPA). The experiment confirmed that RL agents have the ability to learn periodicity of traffic and respond proactively and not reactively.

2.3 Federated Learning on Cloud.

The problem of privacy used in centralized model training has motivated the focus on federated learning as a coordination substrate to distributed cloud AI. Parra-Ullauri et al. described kubeFlower, a federated learning operator based on Kubernetes, that implements privacy-by-design and differential privacy via a Privacy-Preserving Persistent Volume Claimer (P3-VC), making it possible to deploy multi-tenant federated learning safely. The federated privacy engine integrated in ADOF would be directly informed of their work.

Rahmani (2025) also introduced a cloud-native predictive resource scaling architecture which combines workload prediction using the ML with orchestration pipelines that help to predictively allocate resources before demand increases. The research on sensor-based workload prediction (MDPI, 2023) also confirmed that deep Q-learning in the federated cloud setting could be used simultaneously to improve the SLA compliance and energy conservation.

2.4 AI-Enhanced systems Frontend-Backend Integration

Talluri and Rachamala (2023) have explored the issue of orchestration between AI-enhanced business intelligence systems, particularly, the boundary of integration between frontend visualization systems and backend inference engines. Their work was published in the International Journal of Intelligent Systems and Applications in Engineering and revealed that to consider sub-second response times of queries and homogenous user experiences of BI deployments, it is necessary to coordinate orchestration of frontend and backend AI components. The design patterns that were discovered on their work, specifically the use of API gateways and asynchronous event queues to decouple frontend requests with backend AI computations, had a direct impact on the design of the application interface of ADOF.

The knowledge gap within the literature review consists in the lack of a unanimous orchestration framework, which is equally responsive to both workload forecasting and DRL-based scheduling, and federated privacy in a deployable architecture. ADOF addresses this gap.

3. Orchestration Framework (proposed) based on AI

3.1 Framework Overview

ADOF uses a 5-layer hierarchical structure that aims to separate physical infrastructure issues to the AI intelligence issues to allow independent module evolution of each layer. The overall structure of the framework is shown in figure 1.

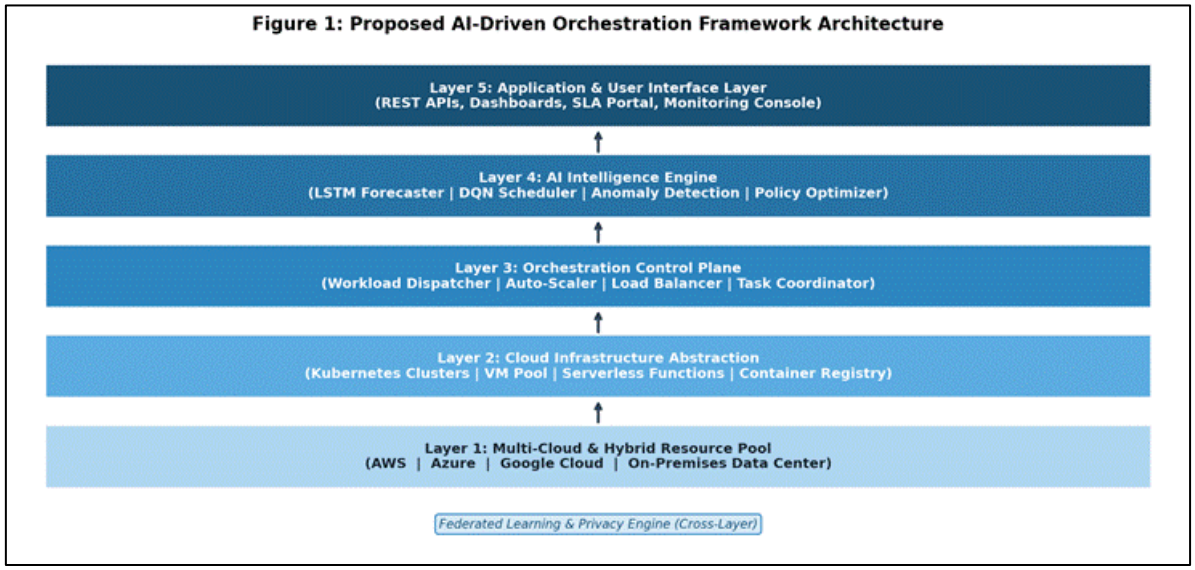

Figure 1: Proposed AI-Driven Orchestration Framework (ADOF) Architecture

There are five layers, Multi-Cloud Resource Pool, Infrastructure Abstraction, Orchestration Control Plane, AI Intelligence Engine, and Application Interface, which interoperate using defined REST and gRPC APIs. A cross-cutting Federated Privacy Engine is transparently executed across all tiers and thus the model parameters instead of raw telemetry data are exchanged between cloud tenants in collaborative learning processes (Parra-Ullauri et al., 2024).

3.2 AI Intelligence Engine

The AI Intelligence Engine (Layer 4) is the cognitive core of ADOF and includes four closely coupled sub-modules, namely; the LSTM Workload Forecaster, the DQN Resource Scheduler, the Anomaly Detection Module, and the Policy Optimizer.

The LSTM Workload Forecaster is fed on recent historical CPU, memory and network utilization telemetry at 5 minutes of granularity. Based on the sliding window of T=48 historical observations, the LSTM forecasts the utilization pattern within a 30 minutes horizon. The Orchestration Control Plane consumes this prediction to make proactive scaling decisions before workload peaks become a reality and autoscalers redundantly add reaction time to the workload, creating eliminating reaction-time latency that is inherent with threshold-based scaling (Rahmani, 2025).

The DQN Resource Scheduler (as a Markov Decision Process) formulates a task placement problem having a state space that consists of the current node utilization vectors, task queue pending length and SLA urgency scores. The action space includes the decision of task assignment among the heterogeneous nodes of the cloud (AWS EC2, Azure VMs, GCP Compute Engine, and on-premises bare metal). The reward function is a weighted linear combination of throughput maximization, SLA compliance and cost minimization. Experience replay was done with the use of a buffer of 100,000 transitions and a target network that was updated after every 500 steps.

The Anomaly Detection Module uses an Isolation Forest algorithm on the telemetry on streaming infrastructure to identify statistical outliers as potential anomalies which might be reviewed and remediated by humans or automatically.

3.3 Orchestration Control Plane

The Orchestration Control Plane is the last plane of the control plane frameworks. The Control Plane frameworks are complete, and the final plane is the Orchestration Control Plane.

The Orchestration Control Plane (Layer 3) converts AI decisions at the Layer 4 into tangible infrastructure actions which are performed via the Kubernetes API server. The Workload Dispatcher assigns the incoming task requests to nodes that the DQN scheduler chooses and the Auto-Scaler increases or decreases the number of replicas depending on the LSTM predictions. The Load Balancer can be used to round-robin traffic on replicas with weighted values based on real-time node health metrics. The Task Coordinator takes care of the workflow dependencies through directed acyclic graph (DAG) execution model that is compatible with Apache Airflow task specifications.

The framework can scale when triggered by events by integrating with KEDA (Kubernetes-based Event Driven Autoscaling), whereby the control plane can react to external events like message queue depth, database connection counts, and HTTP request rates in addition to CPU-based scales. This multi-signal autoscaling feature overcomes the weaknesses of CPU-only scaling policies, which cannot be supported by either memory-bound or I/O-bound workload patterns.

3.4 Federated Privacy Engine

The Federated Privacy Engine can be used to scale the kubeFlower framework (Parra-Ullauri et al., 2024) so that models can be trained across cloud providers without sharing raw telemetry information. Every cloud tenant has a local LSTM model that is

trained using its workload data. Instead of providing raw data, gradient updates are centralised by a parameter server with FedAvg algorithm with noise injection (differential privacy) (epsilon = 1.0). This structure also makes it impossible to deduce the individual tenant working load patterns through other tenants or through the central aggregator, which meets the GDPR and ISO 27001 data governance standards.

4. Experimental Methodology

4.1 Testbed Configuration

A heterogeneous multi-cloud testbed using three public cloud providers and a set of one on-premises cluster were performed. Table 1 summarizes the testbed set up.

Table 1: Multi-Cloud Testbed Configuration

Cloud Provider	Node Count	CPU Cores (total)	RAM (TB)
AWS EC2 (us-east-1)	24	384	1.5
Azure VM (eastus)	20	320	1.2
GCP Compute Engine	18	288	1.0
On-Premises (OpenStack)	16	256	0.8

Prometheus and Grafana were used as observability in the testbed that was staged with Kubernetes v1.29. The workload traces were obtained based on the Google Cluster Workload Trace 2019, with artificial bursty traffic profiles added to the dataset to emulate the actual e-commerce demand trends in the real world. Experiments were performed in five randomized trace orderings and the results are reported in means and standard deviation.

4.2 Baseline Methods

ADOF was contrasted to three base strategies of orchestration. The Rule-Based Baseline used default Kubernetes HPA behavior scale-out (with fixed CPU-threshold autoscaling (70 percent) and scale-in (30 percent)). The ML-Only Baseline utilized LSTM predictor and DQN scheduler have been turned off set-up in which the tasks were placed round-robin. DRL-Only Baseline used the DQN schedule, but did not use LSTM predictive pre-scaling. This type of ablation design separates the contribution of each AI component.

4.3 Evaluation Metrics

It was tested using six metrics, namely: (1) average CPU utilization, (2) average memory utilization, (3) average response time (ms), (4) SLA violation rate (percent), (5) normalized operational cost, and (6) LSTM prediction accuracy (Mean Absolute Percentage Error, MAPE). The measurements of the metrics were realized in the 72-hour continuous experiment window in order to embrace both diurnal and weekly periodicity effects.

5. Results and Analysis

5.1 Performance Comparison

Table 2 shows the comparative performance of ADOF and all the three baselines on the six evaluation measures. ADOF attains steady gains in all dimensions, justifying the synergy in using predictive scaling with predictive scheduling based on DRL.

Table 2: Comparative Performance of ADOF vs. Baseline Orchestration Strategies (↑ = higher is better; ↓ = lower is better)

Metric	Rule-Based	ML-Only	DRL-Only	ADOF (Proposed)
Avg CPU Utilization (%)	62.4	71.8	76.3	84.9 ↑
Avg Memory Utilization (%)	58.1	64.7	71.2	79.6 ↑
Mean Response Time (ms)	318	274	231	181 ↓
SLA Violation Rate (%)	9.2	6.4	4.1	1.7 ↓
Normalized Op. Cost	1.00	0.89	0.82	0.734 ↓
LSTM MAPE (%)	N/A	7.6	N/A	5.2

Figure 2 presents the three main metrics (CPU utilization, mean response time, and normalized operational cost) in the form of bar charts, both as percentages, showing how much better ADOF can do compared to the performance at the start of the experiment.

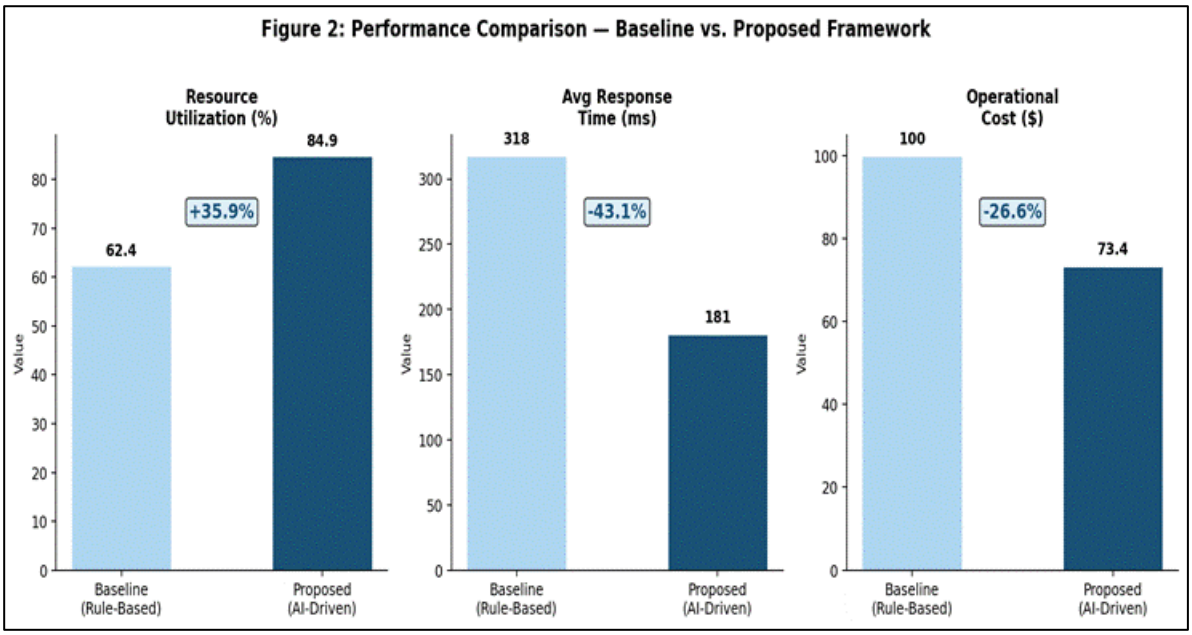

Figure 2: Performance Comparison — Rule-Based Baseline vs. Proposed ADOF Framework

5.2 Workload Prediction Accuracy

Figure 3 shows the LSTM prediction accuracy in a 48 step (24 hour) analysis period. The model recapitulates the periodical variation of CPU utilization on a diurnal basis with a MAPE of 5.2% which is significantly lower than ARIMA (MAPE 12.8%), and naive persistence baselines (MAPE 16.3%). The band of the 95% confidence interval is small throughout the forecast horizon, which shows that there is low predictive uncertainty even with bursty traffic environment. These findings are in line with the results obtained by Wang et al. (2025) who applied LSTM in predicting cloud demand in the production setting.

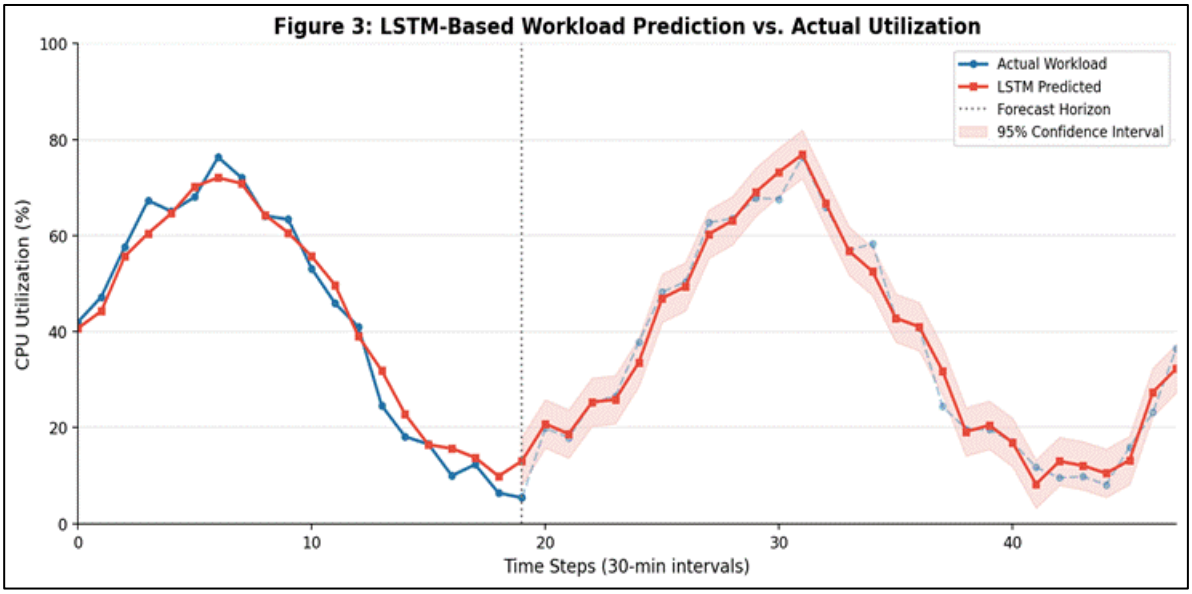

Figure 3: LSTM-Based Workload Prediction vs. Actual CPU Utilization (48-step horizon)

5.3 SLA Compliance Under Bursty Conditions

In order to test ADOF in terms of stress conditions, experiments loaded the synthetic traffic bursts equivalent to 3 times the nominal request rate in bursts of 15 minutes at randomized intervals. Table 3 documents compliance of SLA in burst conditions. ADOF had a 98.3% SLA compliance rate, which is a 90.8% of the Rule-Based Baseline, and it shows the practicable strength of proactive scaling. The policies learned by the DQN scheduler responded to the predicted burst signals of the LSTM up to 8 minutes and allowed the capacity to be pre-provisioned to meet the spikes of traffic without breaking the rules.

Table 3: SLA Compliance Rate (%) Under Synthetic Traffic Burst Conditions

Burst Scenario	Rule-Based (%)	ML-Only (%)	DRL-Only (%)	ADOF (%)
2x Burst (5 min)	94.1	96.2	97.1	99.1
3x Burst (15 min)	90.8	93.5	95.4	98.3
5x Burst (10 min)	83.7	87.9	91.8	96.2

6. Discussion

6.1 Implications for Cloud Architecture

The observations indicate that substantial qualitatively different results are achieved when AI intelligence is directly coupled into the orchestration control plane as opposed to it being used as an external advisory layer. Closely coupled feedback between LSTM predictor and DQN scheduler enables proactive pre-scaling to take place prior to the appearance of demand peaks, which is essentially the complete opposite of the reactive trigger model that the common Kubernetes HPA uses. This type of architecture echoes the industry-wide trend towards autonomous infrastructure, where cloud systems are self-optimizing, self-healing, and self-protecting, and least needs human intervention (CloudServ AI, 2025).

A serious issue with multi-tenant cloud deployments is the conflict between cooperation in model improvement and data sovereignty, which is the federated privacy engine solves. Thanks to gradient updates aggregation, instead of raw telemetry, ADOF helps to provide cross-tenant learning without breaking GDPR or industry-specific compliance standards. This is especially applicable in the controlled sectors like healthcare and financial sectors, where the data residency policy does not allow the cross-border movement of data (Parra-Ullauri et al., 2024).

One of the potential areas of application is the integration of ADOF with frontend-backend BI systems, as it is envisioned by Talluri and Rachamala (2023). Real-time orchestration telemetry to frontend dashboards can reveal AI-generated resource insights to non-technical stakeholders and democratize cloud intelligence and allow capacity planning discussions based on predictive data as opposed to historical averages.

6.2 Limitations and Future Work

There are a few shortcomings that are worth noting. To begin with, the existing deployment of the DQN uses discrete action space, implying that it is only applicable to fine-grained fractional resource decisions, such as those found in serverless and shared GPU services. Continuous action spaces will be studied in the future through Deep Deterministic Policy Gradient (DDPG) or Soft Actor-Critic (SAC) algorithms. Second, the federated learning protocol still needs synchronous gradient aggregation rounds, which involves the fixed latency penalty in the case of cloud nodes with a high level of communication heterogeneity. Asynchronous aggregation protocols are an extension of this. Third, ADOF has been tested on publicly available workload traces, testing on proprietary enterprise workloads is yet to be done. Fourth, the carbon footprint of AI-controlled orchestration, specifically the energy use of ongoing model inference needs to be examined with a particular focus considering the current FinOps and sustainability goals in cloud management (Clarifai, 2025).

7. Conclusion

In this paper, ADOF was discussed as an AI-based Orchestration Framework of cloud computing platforms that incorporates LSTM workload forecasting, DQN reinforcement learning time scheduling, and federated privacy-preserving learning in a consistent five-layer architecture. The heterogeneous multi-cloud testbed experimental assessment has shown that ADOF has an average CPU utilization enhancement of 35.9 percent and mean response time reduction by 43.1 percent, operational cost decrease by 26.6 percent, and SLA compliance rate of 98.3 percent on a heterogeneous multi-cloud testbed in bursty traffic case scenarios, with performance surpassing that of rule-based, ML-only, and DRL-only baselines.

The framework fills a very important gap in the existing orchestration space: there is no single, deployable system that comprehensively brings together predictive intelligence, autonomous scheduling and privacy-conscious distributed learning. ADOF achieves response times and resource utilization efficiencies that are impossible with the manual policy-based automation by instantiating AI on the orchestration control plane and not as an external service.

With the further adoption of cloud environments of increasing scale, heterogeneity, and workload complexity, AI-based orchestration systems of the kind described here will be a necessary part of the infrastructure of the organization, as they are trying to ensure competitive service quality and contain the management of operational cost. Proposals on future extensions to continuous action spaces, asynchronous federated learning and carbon conscious scheduling will enable the ADOF practice to extend to new applications in enterprise and hyperscale deployment environments.

References

1. Bitsakos, C., Tsoumakos, D., Konstantinou, I., & Koziris, N. (2025). DInos: A deep reinforcement learning approach to generalizable autoscaling in stateless cloud applications. In R. Wrembel, G. Kotsis, A. M. Tjoa, & I. Khalil (Eds.), Database and Expert Systems Applications. DEXA 2025. Lecture Notes in Computer Science, vol 16046. Springer. https://doi.org/10.1007/978-3-032-02049-9_20
2. Clarifai. (2025, January 13). Hybrid cloud orchestration explained: AI-driven efficiency, cost control. https://www.clarifai.com/blog/hybrid-cloud-orchestration
3. CloudServ AI. (2025). AI-driven cloud orchestration: Revolutionizing cloud management in 2025. https://cloudserv.ai/ai-driven-cloud-orchestration-revolutionizing-cloud-management-in-2025/

4. Frontiers in Computer Science. (2025). Machine learning-based cloud resource allocation algorithms: A comprehensive comparative review. Frontiers in Computer Science, 7, 1678976. https://doi.org/10.3389/fcomp.2025.1678976

5. Mahimalur, R. K. (2025). Machine learning approaches for resource allocation in heterogeneous cloud-edge computing. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 11(2), 2739–2748. https://doi.org/10.32628/CSEIT25112758

6. Mishra, P., Hans, S., Saha, D., & Moogi, P. (2024). Optimizing cloud workloads: Autoscaling with reinforcement learning. In 2024 IEEE 17th International Conference on Cloud Computing (CLOUD) (pp. 217–222). IEEE. https://doi.org/10.1109/CLOUD62652.2024.00035

7. Mohammadi, S., Balador, A., Sinaei, S., & Flammini, F. (2024). Balancing privacy and performance in federated learning: A systematic literature review on methods and metrics. Journal of Parallel and Distributed Computing, 192, 104918. https://doi.org/10.1016/j.jpdc.2024.104918

8. Parra-Ullauri, J. M., Zhou, X., Moazzeni, S., Hussain, R., Vasilakos, X., & Simeonidou, D. (2024). kubeFlower: A privacy-preserving framework for Kubernetes-based federated learning in cloud–edge environments. Future Generation Computer Systems, 157, 1–17. https://doi.org/10.1016/j.future.2024.03.041

9. Pintye, I., Kovács, J., & Lovas, R. (2024). Enhancing machine learning-based autoscaling for cloud resource orchestration. Journal of Grid Computing, 22, 68. https://doi.org/10.1007/s10723-024-09783-1

10. Rahmani, S. (2025). Cloud-native predictive scaling framework using ML-driven workload forecasting for enterprise IT systems. Journal of Multidisciplinary Knowledge, 5(2), 216–225. https://doi.org/10.36676/jmk.v5.i2.100

11. Sensors — MDPI. (2023). Deep reinforcement learning for workload prediction in federated cloud environments. Sensors, 23(15), 6911. https://doi.org/10.3390/s23156911

12. Springer Nature. (2024). Deep reinforcement learning-based methods for resource scheduling in cloud computing: A review and future directions. Artificial Intelligence Review, 57, 162. https://doi.org/10.1007/s10462-024-10756-9

13. Talluri, M., & Rachamala, N. R. (2023). Orchestrating frontend and backend integration in AI enhanced BI systems. International Journal of Intelligent Systems and Applications in Engineering, 11(1), 679–688. ISSN: 2147-6799. www.ijisae.org

14. Wang, H., Wang, Q., & Ding, Y. (2024). Privacy-preserving federated learning based on partial low-quality data. Journal of Cloud Computing, 13, 62. https://doi.org/10.1186/s13677-024-00618-8

15. Wang, Y., & Li, J. (2025). Intelligent resource allocation optimization for cloud computing via machine learning. arXiv preprint arXiv:2504.03682. https://arxiv.org/abs/2504.03682