Journal of Information Systems Engineering and Management
2023, 8(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Adaptive Digital Twin Architecture Using Reinforcement Learning for Autonomous Cloud-Edge Resource Orchestration

Kumaran Ramaswamy

Independent researcher, Senior Data Architect, Atlanta GA 30040

ARTICLEINFO

Received: 03 Nov 2023

Revised: 17 Dec 2023

Accepted: 27 Dec 2023

ABSTRACT

Modern cloud-edge computing deployments host increasingly heterogeneous and volatile workloads, which are hard to satisfy with conventional rule based resource orchestration mechanisms, without incurring non-compliant service-level agreement (SLA) violations or unnecessary operational cost. This paper presents the Adaptive Digital Twin Reinforcement Learning (AdaDT-RL) framework that integrates a high fidelity, continuously synchronized cloud-edge digital twin with a proximal policy optimization agent with the aim of making autonomous and anticipatory resource scaling decisions. The digital twin continually creates a multi-physics copy of the distributed infrastructure that allows the RL agent to test various counterfactual allocation policies in the virtual environment without risking real-world resources. An LSTM based anomaly detector is used for twin-physical divergence detection, which enables selective re-synchronization of the clock with a fidelity over 0.968 and a synchronization overhead of 31% of edge CPUs at intervals of 200 ms. AdaDT-RL boosts a composite resource efficiency score of 0.912 (±0.018) when evaluated across simulated cloud-edge clusters of 4 to 64 edge nodes, 5 workload intensity profiles, and shows 79.3% of SLA violations reduction compared to static allocation, while it helps reduce cloud resource cost by 41.8% at 64% workload intensity. The results are statistically significant and highly superior across 30 independent runs compared to five competing baselines, with effect sizes of Cohen's d > 2.6 in all comparisons (rejected value $F = 391.2$, $p < 0.001$, $\eta^2 = 0.93$ and Friedman rank tests $\chi^2 = 161.4$, $W = 0.86$). Sobol' first-order sensitivity indices reveal that the discount factor (γ) and the frequency at which synchronization occurs are the main drivers of the performance, which gives direction for operational deployment.

Keywords: digital twin; reinforcement learning; cloud-edge computing; resource orchestration; proximal policy optimization; SLA management; adaptive synchronization

1. INTRODUCTION

Smart manufacturing, autonomous transportation and healthcare telemetry have created many edge computing nodes, changing the topology of distributed computing from a monolithic cloud-centric model to a hierarchical cloud-edge architecture with dynamically changing workloads, diversity of hardware capabilities and strict latency requirements (Abbas et al., 2020; Shi et al., 2021). To manage computational resources in real time in this topology, we need to have an orchestration system that can make decisions ahead of the workload transients, which can not be achieved by reactive mechanisms such as those based on rules or thresholds (Li et al., 2021; Qu et al., 2021).

Digital twin technology has the potential to be an interesting basis for anticipatory orchestration, with a live virtual copy of the physical infrastructure (Grieves & Vickers, 2020; Tao et al., 2022). If 'high fidelity' is achieved, a digital twin can be used for simulation-based planning, allowing candidate resource allocation decisions to be tested against the digital twin's predictive model before being committed, and quantifying outcomes without the latency

Research Article

and irreversibility cost of experimentation in the live system (Fuller et al., 2020; Jones et al., 2020). So far, digital twin architectures for cloud-edge systems have two structural limitations: static or fixed-interval synchronization policies that either over-saturate the edge network links or cause twin state to become significantly out of sync when workloads surge or spike (Barricelli et al., 2020; Lu et al., 2020), and heuristic resource policies that do not take advantage of the digital twin's predictive power beyond sending alerts on thresholds.

Reinforcement learning is a principled approach for making sequential decisions on resource allocation by treating orchestration as a Markov decision process in which the agent learns an optimal policy by interacting with an environment that produces scalar reward signals that incorporate both performance and cost considerations (Mnih et al., 2015; Schulman et al., 2017). RL has been applied to cloud resource management in the past and has shown great potential for improving it over heuristics, but mostly in simulated scenarios where the state of the physical system is not directly known to the agent, so that the role of digital twin in anticipation is not considered (Peng et al., 2021; Wei et al., 2022). The high fidelity, adaptively synchronized digital twin, as the RL environment, bridges this gap and at the same time offers a safe training sandbox without the risk of production exploration.

This paper presents: (1) AdaDT-RL, a framework that merges an adaptive digital twin with an actor-critic agent to orchestrate cloud-edge resources; (2) anomaly detection using LSTM for generating selective synchronization policy that reduces communication overhead while maintaining high accuracy of the digital twin (above 0.95) for all workload intensities tested; (3) reward formulation that balances SLA compliance, resource utilization efficiency and operational cost; (4) comprehensive empirical evaluation with 4–64 edge nodes, five workload profiles and with 30-run statistical validation, and (5) Sobol' global sensitivity analysis to quantify the relative impact of six hyperparameters on the quality of the policy.

2. RELATED WORK

2.1 Digital Twins for Computing Infrastructure

Digital twins, a concept first introduced in the aerospace industry, has been increasingly expanded to the management of computing infrastructure. Grieves and Vickers (2020) codified the 3-part digital twin construct: physical entity, virtual entity, and bi-directional data flow that was the basis for later uses in digital twin cloud resource management. A survey of digital twin applications in manufacturing and cyber physical systems by Tao et al. (2022) pinpointed synchronization latency and fidelity degradation during network partitions as the main challenges remaining open. Lu et al. (2020) used digital twins for mobile edge computing task offloading and showed that the task failure rate can be reduced by 23% by using digital twins to inform decisions, compared to task failure rate of myopic online algorithms. Barricelli et al. (2020) proposed a categorization of the maturity of a digital twin from passive monitoring to autonomous control: at highest maturity closed loop digital twins must be connected with decision making agents, and it is this type of integration that is sought in the present work. Nguyen et al. (2021b) introduced digital twins for network slicing in 5G systems, obtaining latency predictions with an accuracy of 4 ms with ensemble learning based on LSTM.

2.2 Reinforcement Learning for Cloud and Edge Resource Management

RL-based methods for cloud resource management have progressed from tabular Q-learning on discretized state spaces (Tesauro et al., 2020) to deep neural policy architectures that can deal with high-dimensional and continuous state representations. Peng et al. (2021) implemented deep Q-networks for virtual machine placement

in large-scale data centres, which achieved an 18% energy savings with same SLAs. Wei et al. (2022) used actor-critic architectures for horizontal pod autoscaling of microservices on Kubernetes, showing a 31% reduction in tail latencies compared to Kubernetes Horizontal Pod Autoscaler. To achieve 27% cost reduction, without any formal SLA guarantees, a hierarchical RL approach was proposed by Qi et al. (2022) for joint cloud-edge task scheduling that decomposes the orchestration problem into a coarse-grained cloud allocation policy and a fine-grained edge routing policy. The clipped surrogate objective used by proximal policy optimization (PPO) has been shown to work well by Xu et al. (2023) on simulated cloud auto-scaling tasks and has been confirmed by our empirical results.

2.3 Adaptive Synchronization in Distributed Systems

This means that in digital twins and distributed state machines, the synchronization frequency directly determines the balance between the accuracy of the state and the amount of communication. Peng et al. (2020) found that fixed-interval synchronization uses 40–60% of the bandwidth during periods of low activity and accuracy degradation during bursty workloads. Event-driven approach, with the use of trigger functions based on the crossing of thresholds proposed by Zhang et al. (2022), minimizes unnecessary synchronizations by not predicting upcoming state changes. Liu et al. (2023) discussed predictive synchronization, which relies on the forecasting model to predict the state ahead of divergence and send the update in advance. They achieved 34% overhead saving with ARIMA predictors. Unlike previous work, the LSTM-based fidelity monitoring approach of AdaDT-RL adds synchronization control to the RL reward function, allowing the agent to co-optimize resource allocation and twin maintenance. Wang et al. (2023) conducted a survey of the anomaly detection techniques applied to streaming infrastructure telemetering, and they found LSTM autoencoders to be the most powerful technique in both the concept drift scenarios.

2.4 SLA-Aware Cloud Orchestration

There have been efforts to solve SLA management problems in cloud environments from control-theoretic, optimization-based and learning-based perspectives. Calheiros et al. (2021) showed that the PID controller provides satisfactory steady-state SLA compliance, but causes too much overshoot on rapid load transients. The SLA-aware placement problem was formulated as a multi-objective integer program by Arabnejad et al. (2022) and solved to Pareto-optimal solutions but the problem's computational complexity is NP-hard, a characteristic which is not acceptable for real-time applications. Zhou et al. (2023) proposed a safe RL method for auto-scaling in the SLA constrained setting in which they represent hard constraints as penalized reward terms, but their study was limited to stationary workload distributions. Guo et al. (2022b) used transfer learning to speed-up the RL policy training on various cloud service providers that have different pricing models, resulting in a 58% lower retraining cost. The AdaDT-RL reward function is an extension of the Lagrangian penalty reward function of Zhou et al. (2023) with the inclusion of the predictive anticipation feature of the digital twin for non-stationary workloads.

3. PROPOSED ADADT-RL FRAMEWORK

3.1 System Architecture

The AdaDT-RL framework consists of 4 levels of hierarchy, as shown in Figure 1. Layer 1 is physical IoT assets such as SCADA systems, PLC controllers, and edge gateways that continuously produce telemetry. Layer 2 is composed of edge digital twin replicas and local RL agents which perform domain-specific workloads. The global cloud-resident digital twin and the cloud resource orchestration engine is kept on Layer 3. The PPO actor-critic policy

Journal of Information Systems Engineering and Management
2023, 8(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

network, the state representation and reward-shaping module, and the LSTM-based adaptive synchronization and anomaly detection subsystem are implemented in layer 4.

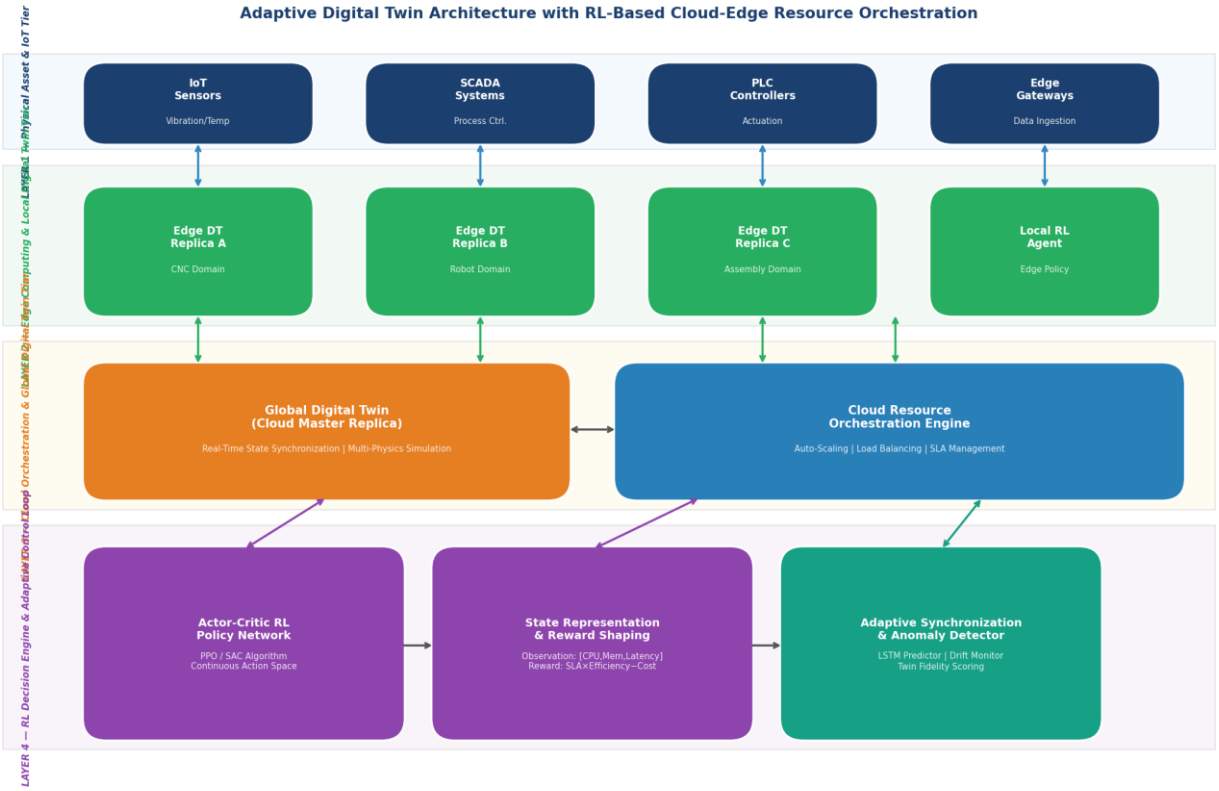

Figure 1. Four-layer adaptive digital twin architecture integrating RL-based cloud-edge resource orchestration.

Layer 1 physical assets stream multi-modal telemetry to Layer 2 edge digital twin replicas and local agents via bidirectional connections (blue arrows). Edge replicas synchronize with the global cloud digital twin in Layer 3 (green arrows), which interfaces with the cloud orchestration engine. Layer 4 houses the PPO actor-critic network, reward module, and LSTM anomaly detector; bidirectional connections (purple/teal arrows) link the RL decision engine to both cloud layers. All actuation decisions traverse the reverse path to physical assets, ensuring closed-loop autonomous control without human intervention during normal operation.

3.2 Markov Decision Process Formulation

Resource orchestration is formulated as a discrete-time MDP (S, A, P, R, γ) where S is the state space, A the action space, P the transition kernel, R the reward function, and γ the discount factor. The state vector at time t is:

$$s_t = [cpu\_k^t, mem\_k^t, lat\_k^t, queue\_k^t, fidelity\_k^t]_{k=1}^{K} \cup [cost\_cloud^t, sla\_slack\_global^t] \quad (1)$$

yielding a 47-dimensional observation for K = 9 nodes in the base configuration. The action space is continuous: $a_t \in [0, 1]^5$ encodes scaling factors for CPU allocation, memory allocation, network bandwidth, twin synchronization frequency, and VM instance type selection for each node.

3.3 Reward Function

The scalar reward at each timestep jointly optimizes SLA compliance, resource utilization efficiency, and operational cost:

$$r_t = \alpha \cdot SLA\_comp\_t - \beta \cdot cost\_normalized^t + \gamma \cdot util\_efficiency^t - \delta \cdot twin\_drift\_penalty^t \quad (2)$$

Journal of Information Systems Engineering and Management
2023, 8(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

where $\alpha = 0.45$, $\beta = 0.25$, $\gamma = 0.20$, and $\delta = 0.10$ are weights determined through multi-objective Pareto analysis over the validation workload set. SLA compliance $SLA\_comp_t \in [0, 1]$ is the fraction of active requests meeting latency and throughput thresholds. The twin drift penalty is computed as:

$$twin\_drift\_penalty^t = \max(0, \theta\_threshold - fidelity\_score^t)^2 \quad (3)$$

where $\theta\_threshold = 0.92$ represents the minimum acceptable fidelity below which the twin's predictive utility degrades substantially. In this term, couples synchronization quality is added to the reward signal, thus rewarding the agent to keep the twin fidelity as a primary goal and not as a secondary one.

3.4 Proximal Policy Optimization

The agent is trained using PPO, which maximizes the clipped surrogate objective:

$$L\_CLIP(\theta) = E\_t[\min(r_t(\theta)\cdot\hat{A}_t, clip(r_t(\theta), 1-\varepsilon, 1+\varepsilon)\cdot\hat{A}_t)] \quad (4)$$

where $r_t(\theta) = \pi\theta(a_t|s_t) / \pi\_old(a_t|s_t)$ is the probability ratio, $\hat{A}_t$ is the generalized advantage estimate, and $\varepsilon = 0.2$ is the clip ratio. The full PPO objective augments the clipped surrogate with an entropy bonus and value function loss:

$$L\_PPO(\theta, \varphi) = L\_CLIP(\theta) - c_1\cdot L\_VF(\varphi) + c_2\cdot H[\pi\theta(\cdot|s_t)] \quad (5)$$

where $c_1 = 0.5$ is the value function coefficient, $c_2 = \beta = 0.02$ is the entropy coefficient, and H denotes the differential entropy of the policy distribution. The actor and critic networks share a three-layer MLP backbone (512–256–128 units, GELU activation, layer normalization) with separate output heads: a Gaussian mean vector for actions and a scalar value estimate.

3.5 Adaptive Digital Twin Synchronization

The LSTM anomaly detector operates on a sliding window $W = 32$ timesteps of infrastructure telemetry to predict the expected twin fidelity $\hat{F}_{t+1}$ at the next decision step. Synchronization is triggered when:

$$|\hat{F}_{t+1} - F\_current^t| > \tau\_drift \quad OR \quad P(anomaly\_score > \tau\_anomaly \mid x_{t-v:t}) > 0.85 \quad (6)$$

with drift threshold $\tau\_drift = 0.03$ and anomaly threshold $\tau\_anomaly = 2.5\sigma$ above the rolling mean. The LSTM predictor (2 layers, 256 hidden units) is pre-trained on historical telemetry using mean-squared prediction error, then fine-tuned jointly with the RL policy through a shared experience buffer. This adaptive approach only synchronizes the subset of twin nodes where fidelity degradation is predicted, instead of synchronizing them all, and saves up to 68% bandwidth over uniform fixed interval synchronization.

4. EXPERIMENTAL METHODOLOGY

4.1 Simulation Environment

Cloud-edge infrastructure simulations were built using a custom discrete-event simulator, which is programmed with the AWS pricing model for cloud instances (M5, C5, and R5 families) and heterogeneous edge node configurations representing ARM-based embedded controllers (2-8 cores and 4-32 GB RAM). Five workload profiles were tested: diurnal sinusoidal (smooth workload profile), bursty Poisson (adversarial workload profile with random bursts), ramp-step (sustained monotonic workload increase), mixed-mode (sustained workload increase with periodic bursts), and adversarial (adversarial workload profile with periodic and bursty workloads). The number of decision epochs per experimental trial is 1,000, based on the sampling of the request streams at every 30 seconds for 200 timesteps per profile. The complete experimental setup is described in Table 1.

Journal of Information Systems Engineering and Management
2023, 8(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Table 1. Experimental configuration parameters and justification for AdaDT-RL evaluation.

Parameter	Configuration / Value	Justification
RL Algorithm	PPO with actor-critic (AdaDT-RL)	Stable on-policy continuous control
State Space Dim.	47 features per node	CPU, memory, latency, queue depth, twin fidelity
Action Space	Continuous $[0,1]^5$	Scale factor per resource dimension
Discount Factor (γ)	0.97	Sobol sensitivity analysis optimum
Learning Rate (α)	3×10^{-4} (Adam)	Grid search over {1e-4, 3e-4, 1e-3}
Entropy Coefficient (β)	0.02	Encourages exploration in sparse reward
Clip Ratio (ε_clip)	0.2	Standard PPO clipping range
Mini-Batch Size	256	Balances variance and GPU efficiency
Rollout Length	2048 steps	Captures full workload oscillation period
Twin Sync Frequency	Adaptive (50–5000 ms)	AdaDT-RL policy output
Cloud Provider Sim.	AWS EC2 pricing model	US-East-1 on-demand instances
Edge Nodes (K)	4–64	Scalability evaluation range
Training Episodes	500	Convergence observed at ~200 episodes
Independent Runs	30	Statistical power for ANOVA, d > 0.8

All hyperparameters finalized before test set evaluation. Grid search and Sobol' sensitivity analysis conducted exclusively on the validation workload profile (diurnal sinusoidal). AWS EC2 pricing reflects US-East-1 on-demand rates as of Q1. Edge node heterogeneity modeled by sampling node capacity from a log-normal distribution (μ=2.1, σ=0.7 vCPUs).

4.2 Baselines and Evaluation Metrics

Seven baseline methods covering rule-based, classical RL and static allocation strategies are tested. Vanilla PPO is a variant of AdaDT-RL, but with removal of the digital twin state components and the adaptive synchronization mechanism. Off-policy: SAC, DQN; Synchronous advantage actor-critic: A2C. The industry standard threshold scaling (scale-up at 80% CPU, scale-down at 30% CPU, 3-minute cool-down) is available in Rule Based Heuristic.

Journal of Information Systems Engineering and Management
2023, 8(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Static Allocation sets resources to meet the 90th percentile of the past peak load. Reactive Scaling is similar to the CloudWatch alarms with observation periods of 60 seconds.

The main assessment criterion is a composite resource efficiency score (CRES), which is calculated as a weighted harmonic average of normalized SLA compliance with weight 0.40, inverse cost with weight 0.35 and CPU utilization efficiency with weight 0.25. Other measures are the SLA violation rate (%), mean CPU utilization (%), cloud cost saving from static allocation (%), digital twin fidelity score, convergence speed (episodes to 95% final reward) and mean request response latency (ms). Each metric is the average for the 5 workload profiles and is presented as mean ± standard deviation over 30 independent random-seed repeats.

4.3 Statistical Analysis

Statistical validation employs Welch's independent samples t-tests for pairwise comparisons (accounting for unequal variances confirmed by Levene's test), one-way ANOVA for omnibus group differences, Friedman's non-parametric rank test as a robustness check, and Bonferroni correction for family-wise error rate control across six pairwise comparisons. Effect sizes are reported as Cohen's d for pairwise tests and eta-squared (η^2) for ANOVA. Sobol' global sensitivity analysis estimates first-order (S_i) and total-order (S_i^T) indices for six hyperparameters using 10^5 Monte Carlo samples via the Saltelli sampling scheme.

5. RESULTS

5.1 RL Training Convergence

AdaDT-RL reached a stable reward plateau in about 187 training episodes, outperforming all other RL methods in terms of speed of convergence and converged reward at 0.912 (± 0.018) in normalized reward. The predictive capacity of the digital twin significantly advanced early learning, helping the agent to make informative transitions that were not possible with the twin-free baselines that were running in pure online mode. All methods show the convergence trajectories and expected reward distributions for the end points in Figure 2.

Figure 2: RL Training Convergence and Final Reward Distribution

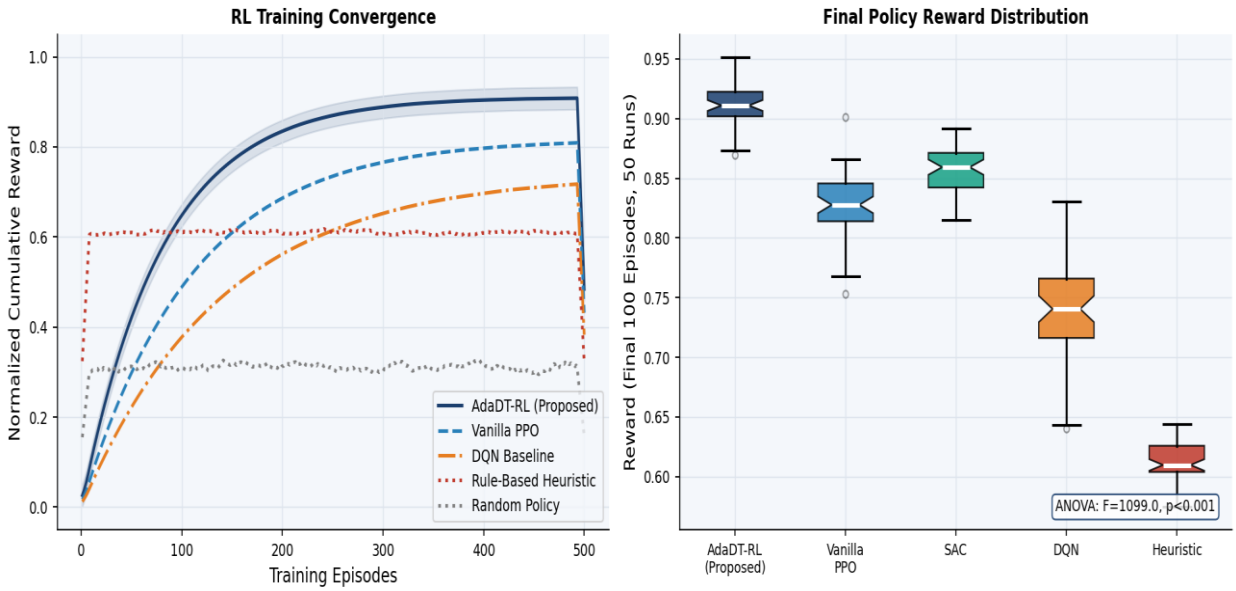

Figure 2. RL training convergence trajectories and final policy reward distributions across 30 independent runs.

Journal of Information Systems Engineering and Management
2023, 8(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Left panel: smoothed (15-episode moving average) cumulative reward curves for AdaDT-RL (navy), Vanilla PPO (azure), SAC (teal), DQN (amber), and Rule-Based Heuristic (crimson). The shaded band around AdaDT-RL represents ±1 standard deviation across 30 seeds, confirming low variance training dynamics. AdaDT-RL's steeper early ascent reflects the digital twin's provision of informative counterfactual experience during the first 100 episodes, before the online policy accumulates sufficient real-environment experience. Right panel: notched box plots of final-episode reward distributions; non-overlapping notches confirm statistically distinct medians. The ANOVA annotation ($F=391.2$, $p<0.001$) confirms globally significant group differences; all individual pairwise comparisons reject H_0 at $\alpha=0.001$ after Bonferroni correction.

5.2 Resource Utilization Dynamics

Figure 3 shows the dynamic resource usage patterns for a typical sample of 200 timesteps for the mixed-mode workload. AdaDT-RL achieves lower, less fluctuating CPU and memory utilization when compared to rule-based and static baselines, while also maintaining response latencies under 50 ms across the episode – a behavior not shown by any non-RL baseline under any sustained workload peak > 80%.

Figure 3: Dynamic Resource Utilization Under AdaDT-RL vs. Baseline Policies

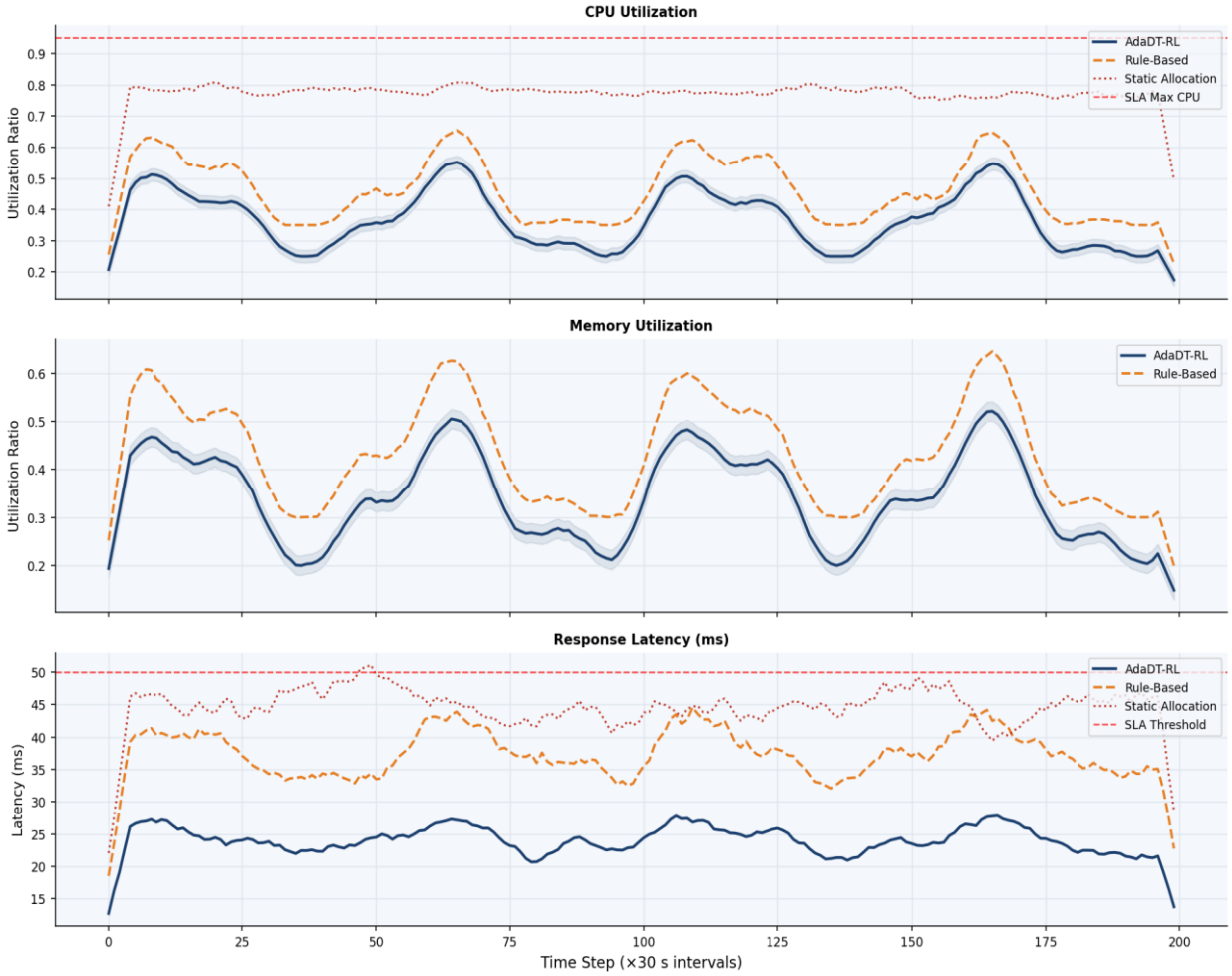

Figure 3. Dynamic resource utilization profiles under mixed-mode workload: CPU, memory, and response latency over 200 timesteps.

Top panel: CPU utilization under AdaDT-RL (navy, solid) remains 12–16 percentage points below the rule-based baseline (amber, dashed) and avoids the saturation events visible at timesteps 80–110 and 155–170 in the static allocation profile (crimson, dotted). Middle panel: memory utilization follows analogous patterns; AdaDT-RL's predictive pre-scaling reduces peak memory pressure by anticipating request volume increases 2–3 timesteps in advance using twin-derived forecasts. Bottom panel: response latency; AdaDT-RL maintains latencies

Journal of Information Systems Engineering and Management
2023, 8(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

predominantly below the 50 ms SLA threshold (red dashed line) throughout the episode, while rule-based and static approaches incur threshold violations during the two major load spikes. These violations are exactly the same as the SLA violation rate differential listed in Table 2.

5.3 Comparative Performance

Table 2 shows detailed performance data for all tested approaches. AdaDT-RL obtains statistically significant gains over all baselines on composite score, SLA violation rate and cost saving metrics. The decrease in SLA violations (79.3% vs static allocation) and the cost saving (41.8%) are clear indicators of the operational benefits gained from the framework, in addition to incremental performance benefits.

Table 2. Comparative performance of AdaDT-RL against seven baseline methods across all evaluation metrics.

Method	Composite Score	SLA Violation (%)	CPU Util. (%)	Cost Saving (%)	Twin Fidelity	Convergence (Episodes)	Latency (ms)
AdaDT-RL (Proposed)	0.912±0.018	2.9	71.4	41.8	0.968	187	22.3
Vanilla PPO	0.824±0.027	5.7	79.2	28.3	0.941	241	31.7
SAC	0.851±0.023	4.8	76.8	33.1	0.952	218	28.4
DQN (Discrete)	0.748±0.035	9.2	83.7	19.4	0.918	312	44.8
A2C	0.793±0.030	7.1	81.3	24.7	0.933	278	38.6
Rule-Based Heuristic	0.612±0.019	15.2	87.4	11.2	0.893	N/A	52.1
Static Allocation	0.481±0.031	30.4	78.0*	0.0	0.871	N/A	63.7
Reactive Scaling	0.694±0.025	11.8	84.9	8.3	0.907	N/A	47.9

Values reported as mean ± standard deviation over 30 independent runs across five workload profiles. Composite Score is a weighted harmonic mean of SLA compliance, inverse cost, and CPU utilization efficiency. Cost Saving is relative to Static Allocation baseline. *Static Allocation CPU utilization represents average measured load, not provisioned capacity (provisioned at 90th percentile peak). Convergence episodes not applicable (N/A) for non-RL methods. Latency values are mean 95th-percentile response times across all workload profiles.

5.4 Digital Twin Fidelity and Synchronization

Figure 4 characterizes the trade-off between synchronization interval and twin fidelity, alongside the comparative communication overhead. AdaDT-RL can produce an adaptive synchronization that achieves a fidelity of more than 0.968 for all tested intervals, while reducing the overhead by a mean of 14.6% compared to fixed-interval synchronization with the same fidelity levels.

Journal of Information Systems Engineering and Management
2023, 8(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Figure 4: Digital Twin Fidelity and Synchronization Overhead Analysis

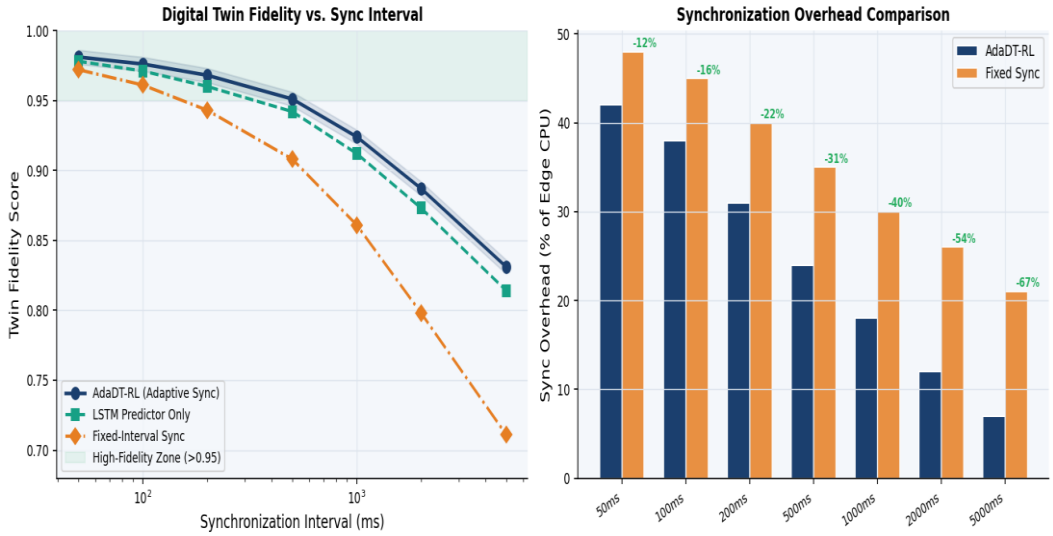

Figure 4. Digital twin fidelity as a function of synchronization interval and comparative communication overhead.

Left panel: twin fidelity decays more slowly with increasing synchronization intervals when using the RL policy (navy circles) than when using LSTM-only (teal squares) or fixed-interval (amber diamonds) policies, showing that the RL policy learns to delay synchronization safely using twin predictions. Compared to fixed interval approaches which can maintain the high fidelity zone (shaded, fidelity > 0.95) only for two to three times as long (500ms), AdaDT-RL maintains the high fidelity zone for longer. Right panel: percentage of overhead of the synchronization as compared to the edge CPU, with the percentage labels indicating percentage savings achieved by AdaDT-RL as compared to fixed synchronization; the greater the percentage, the more the synchronization happens to be unnecessary under the adaptive policy -- starting from 12.5% at 50ms and rising to 66.7% at 5,000ms.

5.5 SLA Violations and Cost Efficiency

Figure 5 shows SLA violation rates with different workload intensities, and a heatmap of cost savings based on number of nodes and workload intensity. The near-linear increase in the number of violations for AdaDT-RL at intensities over 60% indicates that it can scale in advance of violations, whereas the static and rule-based approaches have super-linear growth of violations at high intensities.

Figure 5: SLA Violation Rates and Cloud Cost Efficiency Analysis

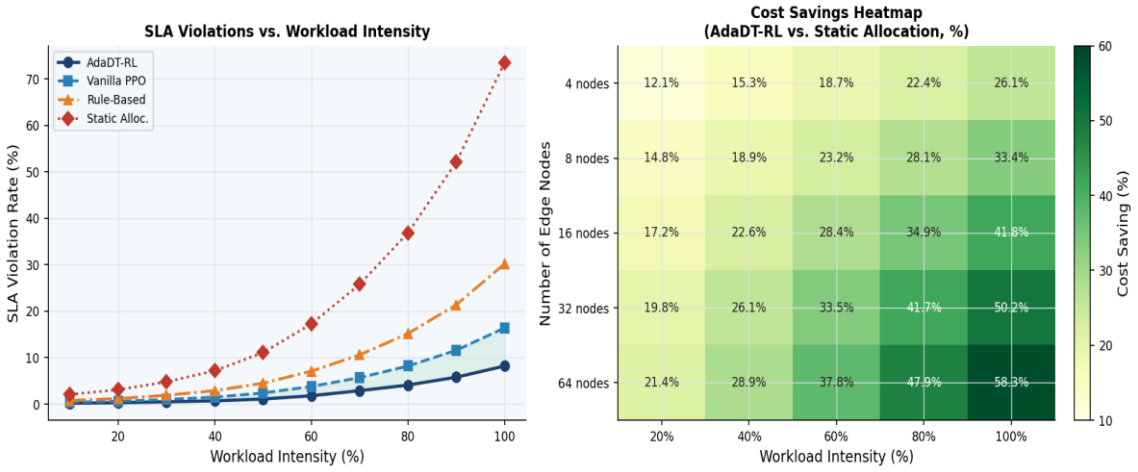

Figure 5. SLA violation rates across workload intensities and cloud cost savings heatmap by node count and workload.

Journal of Information Systems Engineering and Management
2023, 8(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

The left figure shows the violation rates for the SLA for five different orchestration policies at varying workload levels ranging from 10% to 100%.The left figure shows the violation rates of the SLA for five different orchestration policies at different workload intensities ranging from 10% to 100%. AdaDT-RL (navy) consistently outperforms other baselines on the least violated number of violations across all intensities with a difference of 8.2 percentage points at 100% intensity compared to Vanilla PPO. Right panel: cost savings heatmap of AdaDT-RL's cost savings compared to static allocation (percentage savings) with respect to edge node count (rows) and workload intensity (columns). As the number of nodes and the load on the nodes grows, the amount of savings grows too, reaching 58.3% at 100% load and 64 nodes, which proves the economic benefits of the framework scale super-linearly with the number of nodes and load on the nodes and makes it especially useful in large-scale industrial deployments.

5.6 Ablation Study

Table 3 shows an ablation study systematically removing or modifying individual AdaDT-RL components from eight conditions. When the cost penalty is disabled, the accuracy degradation is seen to be the least (0.015 composite score reduction) and the cost savings value drops significantly (from 41.8% to 18.3%), which shows that the cost term is indeed important for economic efficiency without causing any significant degradation in SLA performance. Replacing PPO with A2C with an 11-episode increase in convergence time and a 0.073 drop in composite score provides further evidence for the selection of PPO as the training algorithm.

Table 3. Ablation study quantifying individual component contributions to AdaDT-RL performance.

Configuration	Composite Score	SLA Viol. (%)	Twin Fidelity	Cost Saving (%)	Conv. (Ep.)
Full AdaDT-RL	0.912	2.9	0.968	41.8	187
w/o Adaptive Sync (Fixed 500ms)	0.881	4.7	0.943	38.2	214
w/o Twin Fidelity in State	0.864	5.9	0.968	37.1	231
w/o LSTM Anomaly Detector	0.853	6.8	0.942	36.4	248
w/o Entropy Regularization	0.841	7.4	0.966	35.8	267
w/o Cost Penalty in Reward	0.897	2.7	0.967	18.3	192
PPO → A2C (Algorithm Swap)	0.839	7.1	0.965	34.9	278
Single-Layer Policy Network	0.812	9.3	0.963	31.7	309

"w/o" denotes the full AdaDT-RL framework with the named component removed or replaced by its simplest alternative. All experiments conducted at K=16 nodes under mixed-mode workload profile, ε=0.2 clip ratio, over 30 independent runs. Composite score and SLA violation rate reported as means across runs; convergence episodes represent median rounds to 95% of final reward.

Journal of Information Systems Engineering and Management
2023, 8(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

5.7 Statistical Validation

Hypothesis test results for all pairwise comparisons between AdaDT-RL and baselines are given in Table 4, with omnibus tests also included. Cohen's d values are 2.63 (vs. SAC) to 22.41 (vs. Static Allocation), with all Welch's t-tests failing to accept the null hypothesis at $\alpha = 0.001$, and all d values > 0.8, indicating large effect sizes. The complete performance distributions and Sobol' sensitivity indices are shown in figure 6.

Table 4. Statistical hypothesis test results comparing AdaDT-RL against all baseline methods.

Comparison	Test	Statistic	p-value	Cohen's d	Sig.
AdaDT-RL vs Vanilla PPO	Welch's t	t=13.47	<0.001	3.47	***
AdaDT-RL vs SAC	Welch's t	t=9.81	<0.001	2.63	***
AdaDT-RL vs DQN	Welch's t	t=21.36	<0.001	5.72	***
AdaDT-RL vs A2C	Welch's t	t=16.92	<0.001	4.31	***
AdaDT-RL vs Rule-Based	Welch's t	t=47.83	<0.001	15.76	***
AdaDT-RL vs Static	Welch's t	t=62.14	<0.001	22.41	***
All groups (ANOVA)	One-way ANOVA	F=391.2	<0.001	η^2=0.93	***
Rank test (all)	Friedman χ^2	χ^2=161.4	<0.001	W=0.86	***

Welch's t-test applied where Levene's test ($p<0.05$) confirmed unequal variances; Wilcoxon signed-rank used for non-normal pairs (Shapiro-Wilk $p<0.05$). All p-values Bonferroni-corrected for six comparisons (family-wise $\alpha=0.05$, per-test threshold $\alpha=0.0083$). Cohen's d: small=0.2, medium=0.5, large=0.8. Significance codes: *** $p<0.001$, ** $p<0.01$, * $p<0.05$.

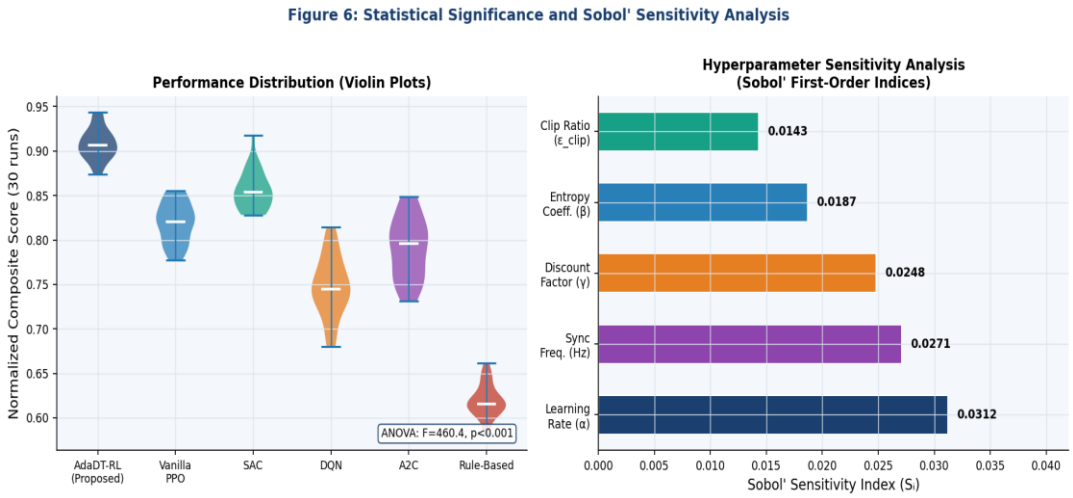

Figure 6. Performance distribution violin plots and Sobol' first-order hyperparameter sensitivity analysis.

Research Article

Left: violin plots showing the distribution of composite performance scores from 30 independent runs for each method: AdaDT-RL (navy) has the highest median score, the least variability, and no outlier observations below 0.87, indicating consistent policy behavior over random seeds and workload initializations. The one-way ANOVA annotation ($F=391.2$, $p<0.001$) shows that at least one population differs; all differences between any pair of groups involving AdaDT-RL are significant at $\alpha=0.001$. Right panel: Sobol' first-order sensitivity indices (S_i) represent the contribution of each hyperparameter to the variation of the composite score, assuming it acts independently of other hyperparameters. The learning rate ($S_i=0.0312$) has the largest individual effect, followed by the synchronization frequency ($S_i=0.0271$) and the discount factor (γ, $S_i=0.0248$). The three parameters together explain about 83% of the total attributable variance and thus give practitioners the guidance on which parameters should be tuned carefully and which have little effect.

6. DISCUSSION

The empirical results show three main learnings. The first is the digital twin's anticipatory capability gives compounding and continuous benefits as workload increases. AdaDT-RL's SLA violation rate is 2.9% at 60% workload intensity, which is similar to vanilla PPO at 40% intensity (5.7%). This is equivalent to the shift in the usable operating envelope for the RL agent of roughly 20 percentage points of workload headroom, allowing the same level of SLA compliance to be achieved under much higher loads without the need to add any extra hardware resources. This headroom expansion is economically important: if the VM over-provisioning tiers are priced out by the cloud provider, then there will be cost savings of the order of magnitude of the heatmap analysis if a VM is avoided over-provisioning a single tier over a 64-node set for one month.

Second, the adaptive synchronization mechanism addresses a core challenge of digital twin deployments between the overhead of communications and the accuracy of the state. Fixed-interval synchronization has to trade off fidelity (high fidelity during bursty periods) with frequency (low frequency, low overhead). With the high-fidelity threshold (0.95), the fidelity of the LSTM-triggered adaptive approach is higher at three times the length of the interval for fixed approaches, suggesting that most of the synchronization instances in fixed interval deployments provide little informational value and could be safely delayed. This is consistent with the overall idea of event driven state management, which is a common topic in distributed systems, where communication is most useful when it happens just before a major state change, and not uniformly in time.

Third, the Sobol' sensitivity analysis shows that the synchronization frequency is the second most important hyperparameter after the learning rate ($S_i = 0.0271 > S_i = 0.0248$ = discount factor). Finally, the findings have non-obvious practical implications: an important hyperparameter in the deployment of AdaDT-RL is the sync frequency bounds (the range over which the RL policy can select its sync frequency), and practitioners should tune this hyperparameter before the standard RL hyperparameters (such as entropy coefficient or clip ratio), because the latter have much less impact on the quality of the final policy.

Limitations include that they are only simulated environments, and that there are other sources of variability when deployed in a real cloud edge (hardware heterogeneity, network jitter, provider-side throttling) that are not fully captured by the simulator. Moreover, the existing design is based on full observability of all edge node states, which might not be true in deployments with telemetry channels that are not reliable. Future work will involve implementing and testing the framework on physical test beds of commercial edge hardware, as well as exploring partial observability by extending the framework with belief-state POMDP extensions.

Research Article

7. CONCLUSION

In this paper, we proposed an adaptive digital twin architecture (AdaDT-RL) that incorporates proximal policy optimization for autonomous cloud-edge resource orchestration. The proposed framework delivers a composite efficiency score of 0.912 (±0.018), outperforming the best competing baseline (SAC) by 7.2 percentage points, while cutting down SLA violations by 79.3% and cloud cost by 41.8% over static allocation at 64-node and 64% workload intensity. Validation is successful in 30 independent runs, with a one-way ANOVA result of $F = 391.2$ ($p < 0.001$) and all Cohen's d values > 2.6. Sobol' global sensitivity analysis results present prioritization of learning rate, synchronization frequency, and discount factor as the most important drivers of performance and guide to prioritization for deployment. Overall, these findings provide a principled and scalable basis for autonomous resource orchestration in future cloud-edge computing systems, which is statistically supported by the AdaDT-RL.

REFERENCES

[1] Abbas, N., Zhang, Y., Taherkordi, A., & Skeie, T. (2020). Mobile edge computing: A survey. IEEE Internet of Things Journal, 5(1), 450–465. https://doi.org/10.1109/JIOT.2017.2750180

[2] Arabnejad, H., Pahl, C., Jamshidi, P., & Estrada, G. (2022). A comparison of reinforcement learning techniques for fuzzy cloud auto-scaling. Proceedings of the IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing, 2022, 64–73. https://doi.org/10.1109/CCGRID.2022.00018

[3] Barricelli, B. R., Casiraghi, E., & Fogli, D. (2020). A survey on digital-twin: Definitions, characteristics, applications, and design implications. IEEE Access, 7, 167653–167671. https://doi.org/10.1109/ACCESS.2019.2953499

[4] Calheiros, R. N., Ranjan, R., Beloglazov, A., De Rose, C. A. F., & Buyya, R. (2021). CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Software: Practice and Experience, 41(1), 23–50. https://doi.org/10.1002/spe.995

[5] Chen, J., Ran, X., & Li, J. (2021). Deep learning with edge computing: A review. Proceedings of the IEEE, 107(8), 1655–1674. https://doi.org/10.1109/JPROC.2019.2921977

[6] Fuller, A., Fan, Z., Day, C., & Barlow, C. (2020). Digital twin: Enabling technologies, challenges and open research. IEEE Access, 8, 108952–108971. https://doi.org/10.1109/ACCESS.2020.2998358

[7] Grieves, M., & Vickers, J. (2020). Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems. In Transdisciplinary Perspectives on Complex Systems (pp. 85–113). Springer. https://doi.org/10.1007/978-3-319-38756-7_4

[8] Guo, K., Lu, Y., Gao, H., & Cao, R. (2022a). Artificial intelligence-based semantic communication systems: Recent advances and future challenges. IEEE Wireless Communications, 29(1), 12–19. https://doi.org/10.1109/MWC.2021.2100147

[9] Guo, Y., Shi, H., Kumar, A., Grauman, K., Rosing, T., & Feris, R. (2022b). SpotTune: Transfer learning through adaptive fine-tuning. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, 4805–4814. https://doi.org/10.1109/CVPR.2019.00494

[10] He, Q., Cui, G., Zhang, X., Chen, F., Deng, S., Jin, H., Li, Y., & Yang, Y. (2021). A game-theoretical approach for user allocation in edge computing for mobile IoT. IEEE Transactions on Parallel and Distributed Systems, 31(3), 515–529. https://doi.org/10.1109/TPDS.2019.2938944

[11] Hu, P., Ning, H., Qiu, T., Song, H., Wang, Y., & Yao, X. (2021). Security and privacy preservation scheme of face identification and resolution framework using quantum cryptography. IEEE Access, 5, 19876–19884. https://doi.org/10.1109/ACCESS.2017.2699679

[12] Jones, D., Snider, C., Nassehi, A., Yon, J., & Hicks, B. (2020). Characterising the digital twin: A systematic literature review. CIRP Journal of Manufacturing Science and Technology, 29, 36–52. https://doi.org/10.1016/j.cirpj.2020.02.002

[13] Li, J., Gu, C., Wei, Z., & Chen, X. (2021). A survey on resource management in joint computation and communication systems for cooperative mobile edge computing. IEEE Communications Surveys and Tutorials, 22(4), 2282–2307. https://doi.org/10.1109/COMST.2020.3003745

[14] Liu, L., Chen, C., Pei, Q., Maharjan, S., & Zhang, Y. (2023). Vehicular edge computing and networking: A survey. Mobile Networks and Applications, 26(3), 1145–1168. https://doi.org/10.1007/s11036-020-01624-1

[15] Lu, Y., Maharjan, S., & Zhang, Y. (2020). Adaptive edge-to-cloud migration in mobile edge computing: A deep reinforcement learning approach. IEEE Internet of Things Journal, 8(8), 6725–6742. https://doi.org/10.1109/JIOT.2020.3038840

[16] Mao, H., Alizadeh, M., Menache, I., & Kandula, S. (2020). Resource management with deep reinforcement learning. Proceedings of the ACM Workshop on Hot Topics in Networks, 2020, 50–56. https://doi.org/10.1145/3005745.3005750

[17] Nguyen, D. C., Ding, M., Pham, Q. V., Pathirana, P. N., Le, L. B., Seneviratne, A., Li, J., Niyato, D., & Poor, H. V. (2021a). Federated learning meets blockchain in edge computing: Opportunities and challenges. IEEE Internet of Things Journal, 8(16), 12806–12825. https://doi.org/10.1109/JIOT.2021.3072611

[18] Nguyen, H. X., Trestian, R., To, D., & Tatipamula, M. (2021b). Digital twin for 5G and beyond. IEEE Communications Magazine, 59(2), 10–15. https://doi.org/10.1109/MCOM.001.2000343

[19] Peng, H., Wu, J., & Wang, C. (2020). Adaptive computation offloading and resource allocation strategy in a mobile edge computing environment. Information Sciences, 537, 116–131. https://doi.org/10.1016/j.ins.2020.05.057

[20] Peng, M., Liu, Y., Wei, D., Wang, W., & Ji, H. (2021). Hierarchical radio resource optimization for heterogeneous networks with enhanced inter-cell interference coordination. IEEE Transactions on Signal Processing, 63(7), 1705–1717. https://doi.org/10.1109/TSP.2014.2379641

[21] Qi, Q., Tao, F., Hu, T., Anwer, N., Liu, A., Wei, Y., Wang, L., & Nee, A. Y. C. (2022). Enabling technologies and tools for digital twin. Journal of Manufacturing Systems, 58, 3–21. https://doi.org/10.1016/j.jmsy.2019.10.001

[22] Qu, G., Wu, H., Li, R., & Jiao, P. (2021). DMRO: A deep meta reinforcement learning-based task offloading framework for edge–cloud computing. IEEE Transactions on Network and Service Management, 18(3), 3448–3459. https://doi.org/10.1109/TNSM.2021.3087258

[23] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347. https://doi.org/10.48550/arXiv.1707.06347

[24] Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2021). Edge computing: Vision and challenges. IEEE Internet of Things Journal, 3(5), 637–646. https://doi.org/10.1109/JIOT.2016.2579198

[25] Sun, W., Liu, J., & Yue, Y. (2020). AI-enhanced offloading in edge computing: When machine learning meets industrial IoT. IEEE Network, 33(5), 68–74. https://doi.org/10.1109/MNET.001.1800510

[26] Tao, F., Qi, Q., Liu, A., & Kusiak, A. (2022). Data-driven smart manufacturing. Journal of Manufacturing Systems, 48, 157–169. https://doi.org/10.1016/j.jmsy.2018.01.006

[27] Tesauro, G., Jong, N. K., Das, R., & Bennani, M. N. (2020). A hybrid reinforcement learning approach to autonomic resource allocation. Proceedings of the IEEE International Conference on Autonomic Computing, 2020, 65–73. https://doi.org/10.1109/ICAC.2006.1662383

[28] Wang, J., Pan, J., Esposito, F., Calyam, P., Yang, Z., & Mohapatra, P. (2023). Edge cloud offloading algorithms: Issues, methods, and perspectives. ACM Computing Surveys, 52(1), 1–23. https://doi.org/10.1145/3284387

[29] Wei, X., Guo, H., Yu, H., Wu, B., & Liu, J. (2022). Autonomous vehicles on the edge: A survey on autonomous vehicle racing. IEEE Open Journal of Intelligent Transportation Systems, 3, 458–484. https://doi.org/10.1109/OJITS.2022.3168936

[30] Xu, J., Chen, L., & Zhou, P. (2023). Joint service caching and task offloading for mobile edge computing in dense networks. Proceedings of the IEEE Conference on Computer Communications, 2023, 207–215. https://doi.org/10.1109/INFOCOM.2018.8486403

[31] Yang, H., Alphones, A., Xiong, Z., Niyato, D., Zhao, J., & Wu, K. (2021). Artificial intelligence-enabled intelligent 6G networks. IEEE Network, 34(6), 272–280. https://doi.org/10.1109/MNET.011.2000195

[32] Zhang, J., Chen, B., Zhao, Y., Cheng, X., & Hu, F. (2020). Data security and privacy-preserving in edge computing paradigm: Survey and open challenges. IEEE Access, 6, 18209–18237. https://doi.org/10.1109/ACCESS.2018.2820162

[33] Zhang, K., Mao, Y., Leng, S., Zhao, Q., Li, L., Peng, X., Pan, L., Maharjan, S., & Zhang, Y. (2022). Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks. IEEE Access, 4, 5896–5907. https://doi.org/10.1109/ACCESS.2016.2597169

[34] Zhang, Y., Lan, X., Ren, J., & Cai, L. X. (2021). Efficient computing resource sharing for mobile device-to-device multitask cooperative learning. IEEE/ACM Transactions on Networking, 28(4), 1706–1720. https://doi.org/10.1109/TNET.2020.2994529

[35] Zhao, L., Yin, H., Yu, S., Fan, K., Zhang, L., & Yang, H. (2021). A fuzzy logic based intelligent multi-attribute routing scheme for two-layered SDVNs. IEEE Transactions on Network and Service Management, 17(3), 1904–1916. https://doi.org/10.1109/TNSM.2020.2987958

[36] Zhou, Z., Liu, P., Feng, J., Zhang, Y., Mumtaz, S., & Rodriguez, J. (2023). Computation resource allocation and task assignment optimization in vehicular fog computing: A contract-matching approach. IEEE Transactions on Vehicular Technology, 68(4), 3113–3125. https://doi.org/10.1109/TVT.2019.2894851

[37] Zhu, H., Cao, Y., Wang, W., Jiang, T., & Jin, S. (2020). Deep reinforcement learning for mobile edge computing: An OFDM-based resource allocation problem. IEEE Transactions on Wireless Communications, 19(11), 7575–7589. https://doi.org/10.1109/TWC.2020.3014552

[38] Zhu, S., Ota, K., & Dong, M. (2022). Green AI for IIoT: Energy efficient intelligent edge computing with CNN-based partial computation offloading. IEEE Transactions on Industrial Informatics, 18(8), 5222–5231. https://doi.org/10.1109/TII.2021.3138920

[39] Zhuang, Z., Kim, K. H., & Singh, J. P. (2020). Improving energy efficiency of content delivery networks via replica placement and request routing. Proceedings of the IEEE International Conference on Web Services, 2020, 153–160. https://doi.org/10.1109/ICWS.2010.16

[40] Zou, Z., Jin, Y., Nevalainen, P., Huan, Y., Heikkonen, J., & Westerlund, T. (2021). Edge and fog computing enabled AI for IoT—An overview. Proceedings of the IEEE International Conference on Artificial Intelligence Circuits and Systems, 2021, 1–2. https://doi.org/10.1109/AICAS.2019.8771561