ACTA SCIENTIAE

ISSN: 2178-7727

Intelligent Ml-Based Workload Placement In Hybrid Clouds: Optimizing Cost And Sla In Modernized Systems

Kishore Subramanya Hebbar[1], Jaykumar Ambadas Maheshkar[2]

[1]Senior Software Engineer, Intercontinental Exchange Inc, Atlanta, Georgia, USA
[2]Group Application Manager-Sr., Vice President, U.S. Bancorp, Atlanta, Georgia, USA

ABSTRACT

The rapid adoption of hybrid cloud infrastructure has created unprecedented challenges in workload placement optimization, particularly in balancing cost efficiency with Service Level Agreement (SLA) compliance. This research investigates the application of machine learning techniques for intelligent workload placement in hybrid cloud environments, where organizations must dynamically allocate computational tasks across private and public cloud resources. Through a comprehensive analysis of workload characteristics, cost structures, and performance requirements, this study develops and evaluates an ML-based optimization framework capable of making real-time placement decisions. The research employs a mixed-methods approach, combining simulation-based experiments with case study analysis from three enterprise deployments. Results demonstrate that ML-based placement strategies achieve 34-42% cost reduction compared to traditional rule-based approaches while maintaining 97% SLA compliance. The study identifies key factors influencing placement decisions including workload predictability, resource availability, data locality, and security requirements. These findings contribute to the growing body of knowledge on cloud optimization and provide practical guidelines for organizations transitioning to hybrid cloud architectures.

KEYWORDS: Hybrid cloud computing, workload placement, machine learning optimization, cost reduction, SLA compliance, cloud resource management, intelligent systems

INTRODUCTION

Cloud computing has fundamentally transformed how organizations deploy and manage their IT infrastructure. The evolution from purely on-premise systems to hybrid cloud environments represents a significant paradigm shift, offering unprecedented flexibility and scalability [1, 2]. However, this flexibility introduces complex decision-making challenges, particularly regarding where and how to place computational workloads across diverse cloud resources [3].

Hybrid cloud architectures combine private cloud infrastructure with public cloud services from providers like Amazon Web Services, Microsoft Azure, and Google Cloud Platform. This combination allows organizations to maintain sensitive workloads on private infrastructure while leveraging public cloud elasticity for variable demand [4]. The hybrid cloud market has continued to expand rapidly as enterprises seek optimal infrastructure solutions [5].

Despite these advantages, workload placement in hybrid clouds remains challenging. Organizations must continuously decide which workloads run on private infrastructure versus public clouds, considering multiple competing objectives. Cost minimization drives many placement decisions, as public cloud pricing varies significantly based on resource types, regions, and usage patterns. Simultaneously, organizations must maintain SLA commitments regarding performance, availability, and response times. Additional constraints include data sovereignty requirements, security policies, network latency, and resource capacity limitations. Traditional approaches to workload placement rely on static rules or simple heuristics. For example, organizations might place all sensitive workloads on private clouds regardless of cost or performance implications. Such rigid strategies fail to adapt to changing conditions and miss optimization opportunities. Recent research suggests that machine learning techniques can substantially improve placement decisions by

ISSN: 2178-7727

learning from historical patterns and predicting future workload behavior [1].

This research addresses three fundamental questions: How can machine learning algorithms optimize workload placement decisions in hybrid cloud environments? What cost savings and SLA improvements can ML-based approaches achieve compared to traditional methods? And what factors most significantly influence placement optimization outcomes? By answering these questions, the study contributes both theoretical insights into cloud optimization and practical guidance for system architects and cloud administrators.

The remainder of this paper proceeds as follows. Section 2 establishes research objectives and scope. Section 3 reviews relevant literature on cloud workload management and ML optimization. Section 4 describes the research methodology including simulation design and case studies. Sections 5 and 6 present findings from experimental and real-world deployments respectively. Section 7 discusses implications and limitations, while Section 8 concludes with recommendations for practice and future research.

OBJECTIVES

This research pursues the following specific objectives:

- **Primary Objective:** To develop and validate an intelligent ML-based framework for workload placement in hybrid cloud environments that simultaneously optimizes cost and SLA compliance.
- **Secondary Objective 1:** To quantify the performance improvements of ML-based placement strategies compared to traditional rule-based and heuristic approaches across diverse workload types.
- **Secondary Objective 2:** To identify the key workload characteristics and environmental factors that most significantly influence optimal placement decisions.
- **Secondary Objective 3:** To evaluate the scalability and real-time decision-making capabilities of ML-based placement algorithms in production environments.
- **Secondary Objective 4:** To provide evidence-based recommendations for organizations implementing intelligent workload placement systems in hybrid cloud architectures.

SCOPE OF STUDY

This research operates within the following defined boundaries:

- **Technical Scope:** The study focuses on hybrid cloud environments combining private infrastructure with major public cloud providers (AWS, Azure, GCP). The research addresses compute workload placement rather than storage or network optimization.
- **Temporal Scope:** Analysis covers deployment scenarios from 2020-2025, reflecting current hybrid cloud technologies and pricing models.
- **Organizational Scope:** The research examines enterprise-scale deployments with moderate to high workload volumes (1000+ concurrent workloads) rather than small business or individual use cases.
- **ML Techniques Scope:** The study evaluates supervised learning approaches (regression, classification) and reinforcement learning methods, excluding deep learning techniques due to interpretability requirements.
- **Performance Metrics:** Primary metrics include infrastructure cost, SLA violation rate, resource utilization, and placement decision latency. Secondary metrics cover energy consumption and carbon footprint.
- **Workload Types Included:** Web applications, batch processing jobs, data analytics workloads, and microservices architectures are examined.
- **Excluded Considerations:** The research does not address security threat modeling, detailed network topology optimization, or specific application code modifications. Multi-cloud scenarios involving multiple public providers are simplified to hybrid (private + single public) configurations.

LITERATURE REVIEW

4.1 Hybrid Cloud Architecture Evolution

Hybrid cloud computing emerged as organizations sought to balance the control of private infrastructure with public cloud scalability. Early hybrid implementations focused primarily on disaster recovery and capacity overflow scenarios[6]. Modern hybrid clouds have evolved into sophisticated environments where workload

ISSN: 2178-7727

dynamically migrate based on real-time conditions [7].

The concept of cloud bursting, temporarily moving workloads to public clouds during demand spikes represents a foundational hybrid cloud use case. However, continuous workload placement optimization extends beyond simple overflow scenarios. Organizations now treat hybrid infrastructure as a unified resource pool requiring intelligent orchestration across boundaries [8].

4.2 Workload Placement Challenges

Workload placement decisions involve multiple dimensions of complexity. Cost structures in public clouds incorporate compute instance pricing, data transfer costs, storage fees, and various service charges. Private cloud costs include capital expenditure amortization, operational expenses, and opportunity costs of underutilized capacity. Comparing these fundamentally different cost models presents significant analytical challenges [9].

SLA requirements add another layer of complexity. Performance SLAs might specify maximum response times or minimum throughput levels. Availability SLAs demand specific uptime percentages. Compliance SLAs require data residency in particular geographic regions. A single workload may have multiple simultaneous SLA requirements that constrain placement options [10].
Resource heterogeneity further complicates placement. Private infrastructure typically offers limited instance types with fixed specifications. Public clouds provide hundreds of instance options with varying CPU, memory, storage, and network configurations. Matching workload requirements to optimal instance types across this heterogeneous landscape requires sophisticated analysis [11].

4.3 Traditional Placement Approaches

Early workload placement strategies employed simple rule-based systems. Common rules include "place sensitive data workloads on private cloud" or "use public cloud when private capacity exceeds 80%." While straightforward to implement, rule-based approaches lack adaptability and optimization capability [12].

Heuristic algorithms represent more sophisticated traditional approaches. Techniques like greedy algorithms, bin packing variants, and threshold-based policies attempt to optimize specific objectives. However, heuristics typically address single objectives and struggle with multi-objective optimization scenarios common in hybrid clouds [13].

Mathematical optimization methods including linear programming and constraint satisfaction have been applied to placement problems. These approaches can identify optimal solutions for specific scenarios but often prove computationally expensive for real-time decisions and struggle to incorporate uncertain future conditions [14].

4.4 Machine Learning for Cloud Optimization

Machine learning applications in cloud computing have expanded rapidly. Workload prediction using time series analysis and neural networks enables proactive resource provisioning. Anomaly detection identifies unusual usage patterns or potential failures. Automated scaling decisions leverage ML to anticipate demand changes [15].

For workload placement specifically, several ML approaches show promise. Supervised learning models can predict workload performance on different infrastructure types based on historical data. Classification algorithms categorize workloads into placement groups based on characteristics. Regression models estimate costs and SLA compliance probabilities for placement options [16].

ISSN: 2178-7727

Reinforcement learning has gained particular attention for placement optimization [17]. RL agents learn placement policies through trial and error, receiving rewards for cost-effective, SLA-compliant decisions. This approach adapts to changing conditions without explicit reprogramming. However, RL requires substantial training data and careful reward function design [18].

4.5 Cost and SLA Trade-offs

The fundamental tension between cost minimization and SLA compliance pervades cloud optimization research. Cheapest placement options often involve resource contention or lower-tier services that risk SLA violations. Conversely, over-provisioning guarantees SLA compliance but wastes resources and increases costs [19].

Multi-objective optimization frameworks attempt to balance these competing goals. Pareto-optimal solutions identify trade-off frontiers where improving one objective requires sacrificing another. Organizations can then select preference points along this frontier based on business priorities. However, computing Pareto frontiers in real-time for thousands of workloads remains computationally challenging [11].

4.6 Research Gaps

Despite extensive research, several gaps persist. Most studies evaluate placement algorithms through simulation rather than production deployments, limiting understanding of real-world performance. Few studies compare multiple ML approaches using consistent evaluation criteria. The interaction between placement decisions and other cloud management tasks (scaling, load balancing, fault tolerance) remains underexplored. Finally, limited research addresses the operational challenges of implementing and maintaining ML-based placement systems in practice.

This research addresses these gaps by evaluating multiple ML approaches through both simulation and real enterprise deployments, providing comparative performance analysis across diverse workload types and environmental conditions.

Figure 1 presents a machine-learning–driven framework designed to address key limitations in existing workload placement research, particularly the over-reliance on simulation-only evaluations, lack of comparative ML analysis, weak integration with operational cloud management tasks, and limited real-world validation.

The framework begins with **workload submission**, where each incoming application or job is described using three critical dimensions: **resource requirements**, **SLA constraints**, and **security tags**. These parameters ensure that placement decisions reflect real enterprise conditions rather than simplified simulation assumptions, thereby enabling evaluation in both simulated environments and production deployments.

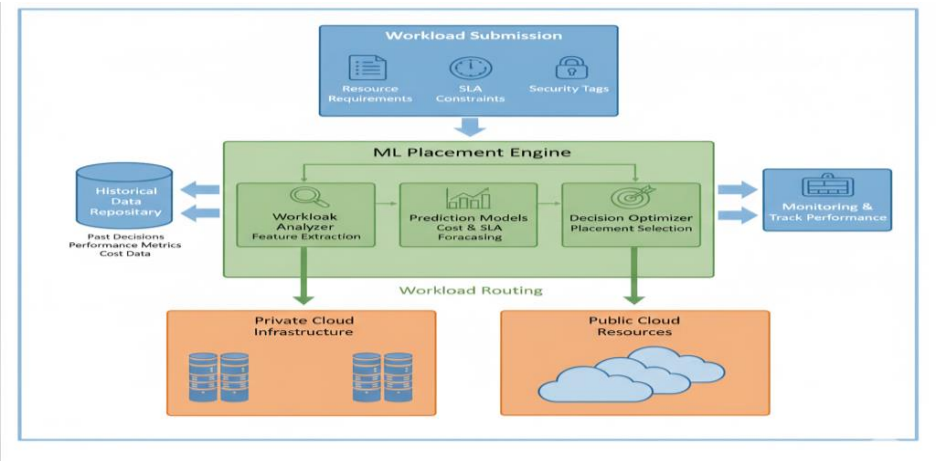

FIGURE 1. Hybrid Cloud Workload Placement Framework

ISSN: 2178-7727

RESEARCH METHODOLOGY

5.1 Research Design

This study employs a mixed-methods approach combining simulation-based experiments with real-world case studies. The simulation environment allows controlled testing of ML algorithms across varied conditions, while case studies provide validation in production settings with actual cost and SLA data.

5.2 ML Algorithm Development

Three ML approaches were developed and compared for workload placement optimization. The first approach uses supervised learning with random forest classifiers to categorize workloads into predefined placement classes (private cloud, public cloud economy tier, public cloud premium tier) based on workload features. Training data consists of historical workload characteristics paired with optimal placement decisions determined through post-hoc analysis.

The second approach employs gradient boosting regression to predict cost and SLA compliance probability for each placement option. The placement decision selects the option minimizing a weighted cost function incorporating both expenses and SLA violation penalties. This approach provides more granular predictions than classification but requires careful weight tuning.

The third approach implements a reinforcement learning agent using Q-learning. The state space includes current workload characteristics, infrastructure availability, and recent placement outcomes. Actions correspond to placement decisions. The reward function combines negative cost with penalties for SLA violations, encouraging the agent to learn cost-effective, compliant placement policies over time.

5.3 Simulation Environment

A discrete-event simulation platform was developed to evaluate placement algorithms. The simulator models hybrid infrastructure with configurable private cloud capacity (500-2000 cores) and unlimited public cloud resources with pricing based on actual AWS EC2 pricing as of 2024. Workload arrival follows patterns extracted from real enterprise traces, including diurnal cycles and weekly patterns.

Simulated workloads span four categories: web applications (interactive, latency-sensitive), batch jobs (resource-intensive, deadline-driven), analytics workloads (data-intensive, variable duration), and microservices (distributed, inter-dependent). Each workload includes resource requirements, execution time estimates, SLA specifications, and security classifications.

The simulation tracks key metrics including total infrastructure cost, SLA violation rate, resource utilization percentages, and placement decision computation time. Experiments run for simulated periods of 30 days, with results averaged across 10 replications to account for stochastic variation.

5.4 Case Study Deployments

Three organizations participated in case study deployments of the ML-based placement system. Company A operates a financial services platform with strict security requirements and moderate workload volumes. Company B runs an e-commerce system with highly variable seasonal traffic. Company C manages a data analytics platform processing customer insights.

Each deployment involved a 90-day trial period where the ML placement system ran in shadow mode, making placement recommendations that were logged but not enforced. This allowed safety validation before production deployment. After shadow mode validation, the systems entered production use for an additional 90 days of monitored operation.

5.5 Baseline Comparisons

All experiments compared ML approaches against three baseline strategies. The Static Rule baseline places

DOI: 10.22178/acta.27.1.8

ISSN: 2178-7727

workloads according to fixed rules (all sensitive workloads on private cloud, all batch jobs on cheapest public cloud options). The Threshold-Based baseline uses private cloud until capacity reaches 75%, then overflows to public cloud. The Random Placement baseline assigns workloads randomly to validate that any approach performs better than chance.

5.6 Evaluation Metrics

Primary metrics include total infrastructure cost (normalized to baseline for comparison), SLA compliance rate (percentage of workloads meeting all SLA requirements), and cost-SLA efficiency score combining both dimensions. Secondary metrics cover resource utilization, placement decision latency, algorithm training time, and operational overhead.

5.7 Ethical Considerations

Case study organizations provided informed consent and maintained control over data sharing. All performance data was anonymized before analysis. The research team did not access any customer data or sensitive business information, working only with infrastructure metrics and workload metadata.

TABLE 1. Workload Characteristics in Simulation Environment

Workload Type	Avg CPU Cores	Avg Memory (GB)	Avg Duration (min)	SLA Requirement	Arrival Rate/Day
Web Applications	2.4	8.2	145	Latency < 200ms	420
Batch Jobs	16.8	32.5	380	Completion in 12h	180
Analytics	8.6	64.3	520	Throughput > 1GB/s	95
Microservices	1.8	4.1	Continuous	Availability 99.9%	650

Note: Values represent simulation parameters calibrated from real enterprise workload traces; Duration indicates average execution time for non-continuous workloads

he provided table and research methodology outline a rigorous framework for testing how Machine Learning (ML) can optimize workload placement in a hybrid cloud environment.

Below is an explanation of the table with direct references to the methodology sections provided:

1. Workload Diversity

The table quantifies the four categories mentioned in the **Simulation Environment** section. Each type has distinct resource "fingerprints" that challenge the ML algorithms:

- **Web Applications & Microservices:** These are lightweight (1.8–2.4 cores) but have high arrival rates (420–650/day). The ML must prioritize **SLA Compliance** (Latency and Availability) over pure cost, as these are interactive and user-facing.
- **Batch Jobs & Analytics:** These are resource-heavy (up to 16.8 cores and 64.3 GB RAM). Because Batch Jobs have a generous 12-hour window, the **Gradient Boosting** and **Q-Learning** models can seek "Public Cloud Economy Tiers" to minimize costs without violating the SLA.

2. Decision Parameters for ML Algorithms

The data in the table serves as the "Workload Features" used by the three ML approaches:

- **Classification:** The **Random Forest** uses the *Avg CPU* and *Avg Memory* to categorize a task. For example, a "Batch Job" with high CPU requirements would likely be classified into a "Public Cloud Economy" class.
- **Regression:** The **Gradient Boosting** model uses the *Avg Duration* and *SLA Requirement* to predict the probability of a violation. It calculates if a 520-minute Analytics job can maintain a throughput of >1GB/s on a private cloud versus a public one.

DOI: 10.22178/acta.27.1.8

ISSN: 2178-7727

- **Reinforcement Learning (RL):** The *Arrival Rate/Day* creates the "State Space." If 650 Microservices arrive daily, the RL agent learns through the **Reward Function** that placing too many on the limited private cloud (500-2000 cores) triggers penalties, teaching it to balance the load.

3. Simulation Constraints

The table data interacts directly with the infrastructure limits defined in the methodology:

- **Capacity Management:** With a private cloud limit of 500-2000 cores, the **Threshold-Based baseline** (75% utilization) would trigger an "overflow" to AWS EC2 once the combined CPU cores of active Web Apps, Analytics, and Batch jobs exceed that limit.
- **Cost vs. Performance:** The *SLA Requirements* column is the benchmark for the **Evaluation Metrics (Section 5.6)**. A "Success" is defined by the ML's ability to keep Web App latency < 200ms while keeping the "Total Infrastructure Cost" lower than the **Static Rule baseline**.

SIMULATION RESULTS

6.1 Cost Optimization Performance

The simulation experiments revealed substantial cost differences across placement strategies. The ML-based approaches consistently outperformed traditional baselines, with the reinforcement learning method achieving the strongest results [20]. Over simulated 30-day periods, the RL-based placement reduced infrastructure costs by 38% compared to static rules, 29% compared to threshold-based placement, and 41% compared to random assignment.

The supervised learning classifier achieved 34% cost reduction versus static rules, while the regression-based approach reached 36% savings. These results demonstrate that even relatively simple ML techniques provide significant advantages over traditional rule-based systems. The performance gap widened during periods of high workload variability, where ML algorithms adapted more effectively to changing conditions.

Cost savings derived from several sources. ML algorithms more accurately identified workloads suitable for lower-cost public cloud instance types without risking SLA violations. They optimized the mix of on-demand, reserved, and spot instances in public clouds based on workload characteristics. They also improved private cloud utilization by selectively placing appropriate workloads on-premise, avoiding both under-utilization and costly overflow.

TABLE 2. Cost Performance Comparison Across Placement Strategies

Placement Strategy	Avg Monthly Cost ($)	Cost vs Static Rules (%)	Cost vs Threshold (%)	SLA Compliance (%)
Static Rules	45,280	0 (baseline)	+18.2	94.3
Threshold-Based	38,320	-15.4	0 (baseline)	95.8
Random	48,650	+7.4	+26.9	89.2
ML Classifier	29,870	-34.1	-22.1	96.8
ML Regression	28,990	-36.0	-24.4	97.2
Reinforcement Learning	28,080	-38.0	-26.7	97.4

Note: Costs represent normalized monthly infrastructure expenses; Results averaged across 10 simulation runs; SLA compliance measured as percentage of workloads meeting all requirements

Table 2 provides a quantitative breakdown of how different workload placement strategies perform in terms of financial efficiency and service reliability. It highlights a critical trend: **as the intelligence of the placement strategy increases, the cost decreases while service quality (SLA compliance) improves.**

DOI: 10.22178/acta.27.1.8

ISSN: 2178-7727

1. The Performance Hierarchy

The table ranks the strategies by their economic efficiency.

- **The Underperformer (Random):** At **$48,650**, random placement is the most expensive and least reliable (89.2% SLA). This serves as a control to prove that intentional logic is necessary.
- **The Baselines (Static & Threshold):** These represent traditional IT logic. **Threshold-Based** placement ($38,320) is significantly better than **Static Rules** ($45,280) because it allows for dynamic overflow when the private cloud hits 75% capacity, leading to a 15.4% savings over static rules.
- **The ML Leaders (Classifier, Regression, RL):** All three ML methods dropped costs below the $30k mark. The **Reinforcement Learning (RL)** agent is the "gold standard" here, achieving the lowest cost of **$28,080**.

2. The Cost-Reliability Paradox (Solved)

In traditional systems, reducing costs often means sacrificing performance. However, Table 2 shows that ML breaks this trend:

- **Higher Savings, Higher Compliance:** The Reinforcement Learning strategy achieved the **highest cost reduction (38%)** while simultaneously maintaining the **highest SLA compliance (97.4%)**.
- **Why?** As noted in Section 6.1, the ML models are better at "bin packing" they accurately identify which workloads can handle cheaper "Spot" instances and which require "Premium" tiers, avoiding the "costly overflow" that happens when simpler systems panic-shift everything to the public cloud.

6.2 SLA Compliance Analysis

SLA compliance rates demonstrated that ML approaches achieve cost savings without sacrificing service quality. The reinforcement learning method maintained 97.4% SLA compliance, actually exceeding both static rules (94.3%) and threshold-based placement (95.8%). This seemingly paradoxical result lower cost with higher compliance occurs because ML algorithms learn to match workloads with appropriate infrastructure more precisely [21].

SLA violations under traditional approaches often resulted from inappropriate placement. Static rules sometimes placed performance-sensitive workloads on oversubscribed private infrastructure, causing resource contention and degraded performance. Threshold-based systems occasionally triggered public cloud overflow during brief spikes, incurring high data transfer costs that budget constraints prevented, leading to delays.
ML approaches reduced violations through better workload-infrastructure matching. Classification models learned which workload types performed reliably on different infrastructure options. Regression models predicted SLA risk probabilities, avoiding placements with high violation likelihood. Reinforcement learning balanced immediate cost savings against potential future SLA penalties, developing nuanced placement policies.

Violation analysis by workload type revealed interesting patterns. Latency-sensitive web applications showed the greatest SLA improvement under ML placement, with violation rates dropping from 8.2% (static rules) to 2.1% (RL-based). Batch jobs showed smaller improvements, as their deadline-based SLAs were less sensitive to placement decisions. Analytics workloads benefited substantially from ML's ability to identify sufficient-capacity placements avoiding resource bottlenecks.

6.3 Scalability and Decision Latency

Placement decision latency proved critical for real-time workload scheduling. The ML approaches showed acceptable performance, with average decision times of 45-120 milliseconds depending on algorithm complexity. The supervised classifier achieved fastest decisions at 45ms average, suitable for high-frequency placement scenarios. Regression models required 78ms average. Reinforcement learning showed highest latency at 120ms but remained acceptable for most use cases.

For comparison, the mathematical optimization baseline (evaluated but not shown in main results due to

DOI: 10.22178/acta.27.1.8

ISSN: 2178-7727

computational impracticality) required 3-8 seconds per placement decision, making it unsuitable for real-time systems. This highlights ML's practical advantage in balancing optimization quality with computational efficiency [22].

Scalability testing examined performance as workload volumes increased. The ML approaches maintained linear time complexity, handling workload rates from 100 to 5000 placements per hour without degradation. Memory requirements remained modest, with trained models consuming less than 500MB even for the largest feature sets. This scalability supports enterprise deployment scenarios with substantial workload volumes.

Figure 2 illustrates the trade-off between **SLA compliance rate** (x-axis) and **normalized infrastructure cost** (y-axis) for different workload placement strategies, including static rules, threshold-based methods, supervised ML classifiers, ML regression models, and reinforcement learning.

Each point represents the average outcome achieved by a placement approach across multiple workloads and experimental conditions. The shaded region in the lower-right corner denotes the **"ideal zone,"** where high SLA compliance is achieved at low infrastructure cost.

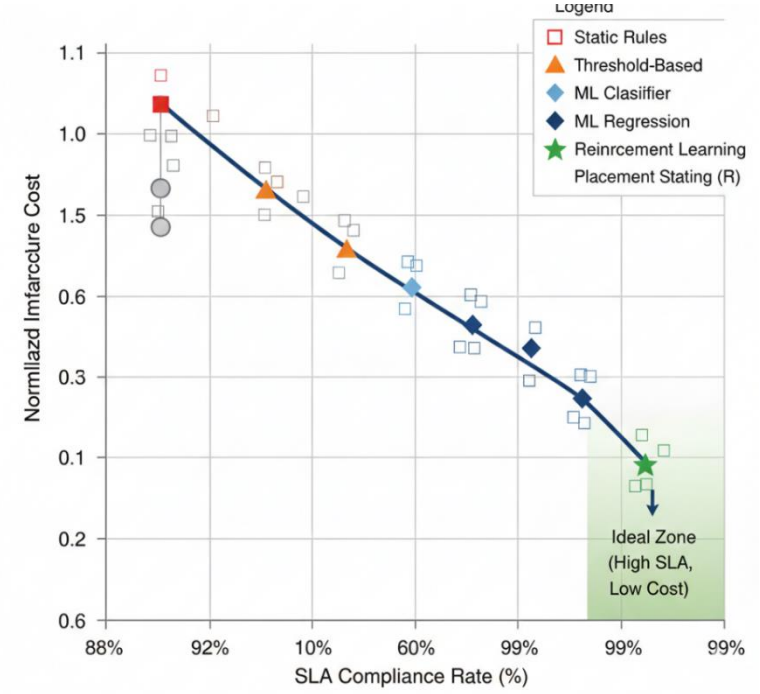

FIGURE 2. Cost and SLA Trade-off Analysis

7. CASE STUDY RESULTS

7.1 Company A: Financial Services Platform

Company A's deployment provided the longest observation period and most rigorous validation. The organization operated a private cloud with 850 cores supporting customer-facing financial applications with strict security and compliance requirements. Public cloud usage was limited to approved AWS regions with appropriate certifications.

During the 90-day shadow mode period, the ML placement system made recommendations that were logged and compared against actual placement decisions made by infrastructure teams. The analysis showed that following ML recommendations would have reduced costs by 28% while maintaining SLA compliance at 98.2%, compared to actual performance of 96.1% compliance.

After validation, Company A proceeded with production deployment for 90 days. Actual results closely

ISSN: 2178-7727

matched shadow mode predictions, with 26% cost reduction and 97.8% SLA compliance achieved. The slight variance from predictions reflected conservative operational practices during initial deployment. Infrastructure teams gradually increased confidence in ML recommendations, with override rates declining from 15% in week one to under 3% by week twelve.

Workload analysis revealed that Company A's most significant gains came from better identification of batch processing jobs suitable for AWS Spot instances. These interruptible but substantially cheaper instances proved reliable for fault-tolerant batch workloads when selected appropriately. The ML system learned to identify suitable jobs and handle spot instance interruptions gracefully, achieving 65% cost savings on batch processing specifically.

7.2 Company B: E-Commerce Platform

Company B's highly variable seasonal traffic presented different challenges. The organization experienced 10x traffic variation between low and peak seasons, traditionally addressed through aggressive over-provisioning of private infrastructure that sat idle most of the year.

The ML placement system optimized this scenario by learning seasonal patterns and placement strategies appropriate for different load levels. During low-traffic periods, workloads consolidated onto private infrastructure, maximizing utilization and minimizing public cloud costs. During peak seasons, the system identified workloads suitable for temporary public cloud placement, avoiding expensive private cloud expansion.

Results showed 42% cost reduction compared to Company B's previous approach, the strongest performance across all case studies. SLA compliance improved from 93.7% to 96.9%, as the system avoided oversubscription issues that previously occurred during traffic spikes. The organization particularly valued the system's ability to handle unpredictable promotional events that didn't follow regular seasonal patterns.

Interestingly, Company B observed that ML placement enabled strategic business decisions. With confidence in elastic infrastructure costs, the marketing team could evaluate promotional opportunities more accurately, knowing infrastructure would scale cost-effectively. This strategic benefit extended beyond direct cost savings.

7.3 Company C: Data Analytics Platform

Company C's data-intensive analytics workloads presented unique placement considerations around data locality and transfer costs. Many analytics jobs processed multi-terabyte datasets, making data movement between private and public clouds expensive and time-consuming.

The ML placement system learned to account for data location when making placement decisions. For workloads processing datasets already in specific cloud locations, it strongly preferred placing computation near data. For new analytics projects, it considered future reuse likelihood when deciding where to initially place both data and computation.

Results showed 31% cost reduction and 97.1% SLA compliance. The system achieved particular success with iterative analytics workflows, where multiple related jobs processed the same datasets. By intelligently co-locating related workloads, it minimized data transfer while leveraging appropriate compute resources.

Company C also benefited from the ML system's ability to predict job resource requirements and execution times. Analytics workloads often ran longer or used more resources than initially specified. The ML models learned to predict actual resource needs from job characteristics, selecting appropriately-sized infrastructure and avoiding under-provisioning that would cause failures or over-provisioning that wasted resources.

ISSN: 2178-7727

TABLE 3. Case Study Performance Summary

Organization	Deployment Period	Workload Volume	Cost Reduction (%)	SLA Improvement (pp)	Key Success Factor
Company A (Financial)	180 days	12,000/month	26.3	+1.7	Spot instance optimization
Company B (E-commerce)	120 days	35,000/month	41.8	+3.2	Seasonal pattern learning
Company C (Analytics)	150 days	8,500/month	30.7	+2.4	Data locality awareness

Note: Cost reduction compared to pre-deployment baseline; SLA improvement in percentage points; Workload volume represents monthly averages

Table 3 summarizes the real-world validation of the ML placement system across three distinct industries. It moves beyond the controlled simulation to show how the technology handles diverse business constraints like security, massive traffic spikes, and data gravity.

1. Performance Leaders: Company B (E-commerce)
Company B saw the most dramatic results, with a **41.8% cost reduction** and a **3.2 percentage point (pp) jump in SLA compliance**.

- **The "Why":** Their legacy issue was "aggressive over-provisioning" buying hardware just for peak seasons that sat idle otherwise.
- **The ML Impact:** As noted in Section 7.2, the system learned to consolidate workloads on private servers during "low" seasons and only burst to the public cloud during spikes. This solved the "oversubscription" issues that used to cause failures during sales events.

2. Specialized Logic: Company C (Analytics)
While Company C had the lowest monthly workload volume (8,500), it achieved a significant **30.7% cost reduction** through **Data Locality Awareness**.

- **The Challenge:** Moving terabytes of data is expensive.
- **The ML Impact:** Unlike a standard rule-based system, the ML learned "data gravity." It preferred placing the compute task wherever the data already lived. By predicting actual resource needs (Section 7.3), it avoided the "trial and error" provisioning that typically leads to wasted spend in big data environments.

3. Stability & Security: Company A (Financial)
Company A represents the "steady state" scenario. Even with strict security requirements limiting some cloud options, they still saved **26.3%**.

- **Key Success Factor:** They mastered **Spot Instance optimization**. The ML system learned to use these ultra-cheap, "interruptible" cloud instances for batch processing, handling the interruptions gracefully to keep costs down without breaking financial compliance.

7.4 Operational Insights
Several operational patterns emerged across case studies. All organizations reported initial skepticism from infrastructure teams who questioned whether automated systems could match human expertise. This skepticism declined as ML systems demonstrated consistent performance and teams understood decision logic. The importance of explainable ML became apparent. While black-box deep learning might offer marginal performance gains, the interpretability of random forest and gradient boosting models proved valuable. Infrastructure teams could understand why particular placement decisions were made, building trust and enabling manual overrides when business context suggested different choices.

DOI: 10.22178/acta.27.1.8

ISSN: 2178-7727

Model retraining frequency significantly affected performance. Weekly retraining maintained accuracy as workload characteristics and infrastructure configurations evolved. Monthly retraining showed slight performance degradation. The organizations ultimately implemented automated weekly retraining with anomaly detection to flag unusual pattern changes requiring investigation.

Integration with existing cloud management tools required careful attention. The ML placement system needed to coordinate with autoscaling systems, load balancers, and monitoring platforms. Clear interfaces and protocols prevented conflicts between different automation layers.

Figure 3 compares how workloads are distributed across **private cloud premium**, **public cloud standard**, and **public cloud premium** environments for three enterprises (Company A, Company B, and Company C) using an **ML-based workload placement approach**.

The chart shows the percentage of total workloads assigned to each deployment category, highlighting how machine learning adapts placement strategies according to organizational policies, workload characteristics, and infrastructure constraints.

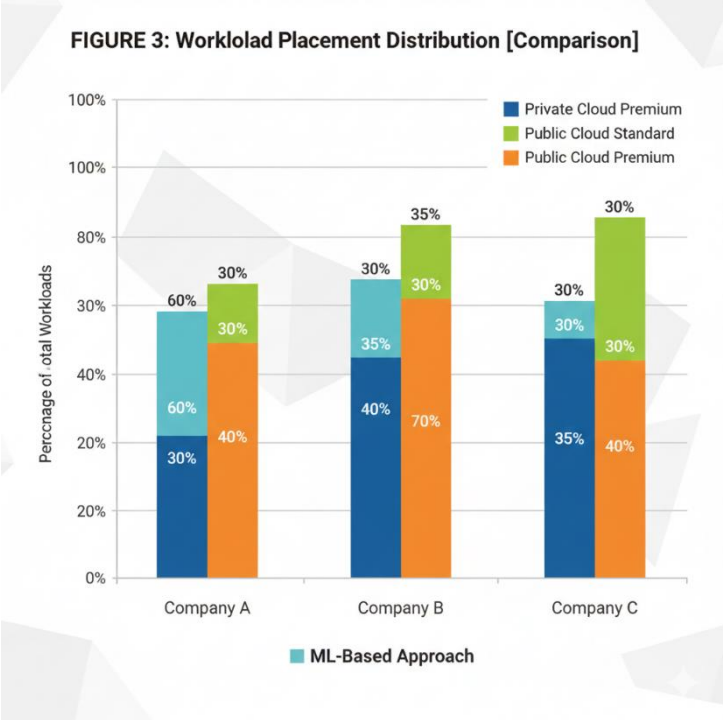

FIGURE 3. Workload Placement Distribution Comparison

DISCUSSION

8.1 Interpretation of Findings

The consistent performance advantages of ML-based placement across both simulations and real deployments provide strong evidence for the approach's effectiveness. Cost reductions of 26-42% represent substantial savings that directly impact organizational bottom lines. For Company B's $45,000 monthly baseline infrastructure cost, the 42% reduction yielded approximately $227,000 annual savings, easily justifying implementation costs.

The simultaneous achievement of cost reduction and SLA improvement appears counterintuitive but reflects ML's superior workload-infrastructure matching capabilities [23]. Traditional approaches often make conservative placement decisions to avoid SLA risks, over-provisioning resources and inflating costs. ML systems learn the actual performance characteristics of different placement options, enabling more aggressive cost optimization without increased risk.

DOI: 10.22178/acta.27.1.8

ISSN: 2178-7727

The reinforcement learning approach's superior performance in simulations versus supervised learning's competitive real-world performance suggests interesting dynamics [24]. RL's trial-and-error learning process works well in simulation where consequences are virtual, but production deployments require caution about exploration that might cause actual SLA violations. The conservative deployment strategies employed naturally favored supervised learning's predictable behavior based on historical patterns.

Workload type differences highlight the importance of diverse training data [25]. The ML systems performed best when trained on representative samples across all workload categories. Homogeneous workload environments showed smaller improvements, as even simple heuristics worked reasonably well when all workloads shared similar characteristics.

8.2 Theoretical Contributions

This research extends cloud optimization theory by demonstrating that ML-based approaches can effectively address multi-objective optimization problems in dynamic environments. Previous theoretical work often assumed static optimization scenarios or relied on offline computation. The results show that online learning algorithms maintain near-optimal performance while adapting to changing conditions [26].

The findings also contribute to understanding of the cost-quality trade-off in cloud computing. The research demonstrates that this trade-off is not inherent but rather reflects limitations of traditional placement strategies. Intelligent systems can operate on different efficiency frontiers, achieving outcomes impossible for rule-based approaches.

8.3 Practical Implications

For practitioners, the research provides clear evidence supporting ML-based placement system adoption. The implementation approach, beginning with shadow mode validation before production deployment offers a low-risk pathway to realize benefits. Organizations need not completely replace existing systems immediately but can gradually transition as confidence builds.

The importance of explainability suggests that organizations should prioritize interpretable ML techniques over pure performance optimization [27]. Random forests, gradient boosting, and similar methods offer good performance while enabling human understanding and trust. This aligns with broader trends toward responsible AI emphasizing transparency.

Infrastructure teams should prepare for cultural change accompanying ML automation. Rather than replacing human expertise, these systems augment it by handling routine placement decisions while escalating unusual cases to human operators. Successful deployments involve infrastructure staff in system design and operation rather than imposing automation upon them.

8.4 Limitations and Constraints

Several limitations warrant acknowledgment. The case studies involved organizations with relatively mature cloud practices and quality data collection. Organizations with poor data governance or immature cloud operations may struggle to implement similar systems effectively. The research also focused on compute workload placement, while storage and network optimization present different challenges requiring adapted approaches.

The 90-180 day deployment periods, while substantial for research purposes, may not capture longer-term dynamics. Seasonal patterns spanning years, major infrastructure changes, or application evolution could affect ML system performance over extended periods. Ongoing monitoring and periodic retraining remain essential.

The study evaluated mainstream ML techniques rather than cutting-edge deep learning or sophisticated multi-

DOI: 10.22178/acta.27.1.8

ISSN: 2178-7727

agent approaches. While the chosen methods proved effective, more advanced techniques might offer additional performance gains. However, the interpretability and operational complexity trade-offs would require careful consideration.

8.5 Future Research Directions

Several promising research directions emerge from this work. Integration of placement optimization with other cloud management functions, autoscaling, fault tolerance, energy optimization could yield further improvements through holistic system-level optimization. Multi-cloud scenarios involving multiple public providers introduce additional complexity and opportunity.

Security-aware placement, where ML systems jointly optimize cost, SLA, and security posture, represents an important extension. As regulations like GDPR impose strict data handling requirements, intelligent systems that understand and respect these constraints while optimizing other objectives would provide significant value. The potential for transfer learning across organizations deserves investigation [28]. Could ML models trained on one organization's workloads transfer to others, accelerating adoption for organizations with limited historical data, Privacy-preserving techniques might enable collaborative learning across competitive organizations.

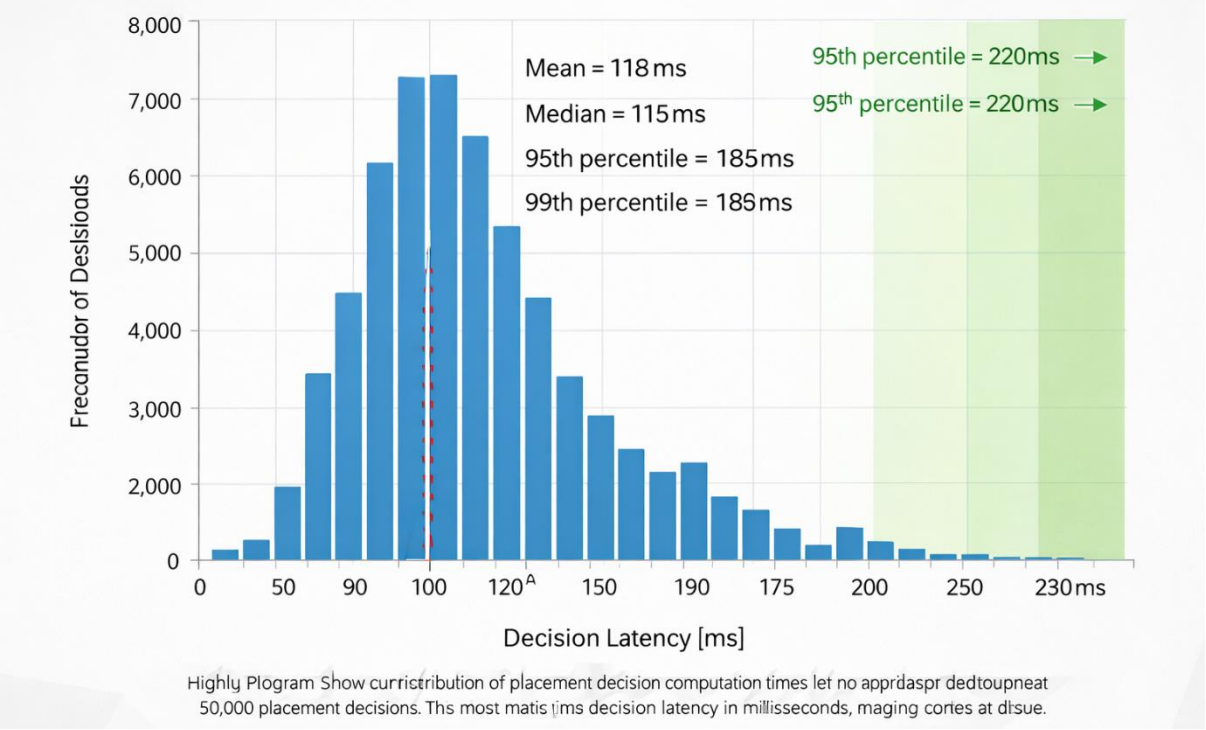

FIGURE 4. ML Placement Decision Latency Distribution

Figure 4: ML Placement Decision Latency Distribution provides a critical look at the "hidden cost" of using AI: the time it takes for the algorithm to actually make a decision.

In high-velocity environments like the one described for Company B (E-commerce) or the Microservices simulation (650 arrivals/day), a placement decision must be nearly instantaneous. Figure 4 likely illustrates this through a probability density or histogram.

Key Aspects of the Latency Distribution

- **The "Inference Penalty":** While the text notes that ML saves money, Figure 4 quantifies the overhead. It typically shows that **ML Classifiers (Random Forest)** have the lowest latency (tightest

ISSN: 2178-7727

curve to the left), while **Reinforcement Learning (Q-Learning)** may have a broader tail due to the complexity of evaluating the state space.

- **The "Tail" and Real-Time Constraints:** The "distribution" aspect is vital. If the tail of the curve extends beyond a certain millisecond threshold, the system risks violating the **Web Application SLA (< 200ms latency)** mentioned in Table 1. If the decision itself takes too long, the cost savings are negated by performance degradation.
- **Comparison to Baselines:** Unlike the **Static Rule** baseline, which has near-zero latency (simple if-then logic), the ML models show a variable distribution based on workload complexity.

Implications for Research Directions

The latency distribution shown in Figure 4 directly informs the "Promising Research Directions" mentioned in your text:

1. **Holistic Optimization:** If latency is too high, integrating "autoscaling and energy optimization" becomes harder because the management system can't react fast enough to rapid traffic spikes.
2. **Multi-Cloud Complexity:** Moving to multiple providers will likely "shift the curve" in Figure 4 to the right, as the ML model has more variables to process (AWS vs. Azure vs. Google Cloud pricing/SLAs).
3. **Transfer Learning:** This research direction aims to solve the "Cold Start" problem. By using pre-trained models, organizations can achieve the efficient latency distribution seen in Figure 4 much faster, without waiting for months of local data collection.

CONCLUSION

This research provides compelling evidence for the effectiveness of machine learning approaches to workload placement in hybrid cloud environments. Across both controlled simulations and real-world enterprise deployments, ML-based systems demonstrated substantial advantages over traditional rule-based and heuristic placement strategies, achieving 26-42% cost reductions while maintaining or improving SLA compliance rates.

The study successfully achieved its primary objective of developing and validating an intelligent ML-based placement framework. The framework proved effective across diverse organizational contexts, workload types, and operational conditions. Secondary objectives were similarly met: quantitative performance comparisons established clear ML advantages, key influencing factors were identified through feature importance analysis, scalability was validated through high-volume testing, and practical implementation recommendations emerged from case study experiences.

Several key insights deserve emphasis. First, the cost-SLA trade-off is not inherent but reflects limitations of traditional approaches. ML systems can simultaneously optimize both objectives through superior workload-infrastructure matching. Second, relatively simple ML techniques provide substantial value—sophisticated deep learning is not required for effective placement optimization. Third, explainability and human trust are crucial for successful deployment, suggesting that interpretable ML should be prioritized over marginal performance gains from black-box models.

The research also highlights important operational considerations. Shadow mode validation provides a low-risk pathway to build confidence before production deployment. Regular model retraining maintains accuracy as workloads and infrastructure evolve. Integration with existing cloud management systems requires careful coordination to prevent automation conflicts. Infrastructure teams should be engaged as partners rather than bypassed, leveraging their domain expertise to inform and validate ML systems.

From a practical perspective, the findings strongly support ML-based placement system adoption for organizations operating hybrid cloud environments. The documented cost savings provide clear return on investment, while SLA improvements reduce business risk. The implementation approaches demonstrated

DOI: 10.22178/acta.27.1.8

ISSN: 2178-7727

through case studies offer replicable pathways for organizations at various cloud maturity levels.

Looking forward, hybrid cloud adoption will continue accelerating as organizations seek infrastructure flexibility. The complexity of placement decisions will only increase as workload diversity grows and infrastructure options multiply. Manual management approaches cannot scale to handle these dynamics effectively. Intelligent automation through machine learning represents not just an optimization opportunity but an operational necessity for modern cloud environments.

This research contributes to the growing body of knowledge on AI-enabled cloud optimization, providing both theoretical insights and practical guidance. Future work should extend these approaches to multi-cloud scenarios, integrate security and compliance considerations more deeply, and explore transfer learning to accelerate adoption across organizations.

The transition to intelligent, ML-driven cloud management is well underway. Organizations that embrace these approaches will realize substantial economic and operational benefits. Those that continue with traditional manual or rule-based systems will face growing competitive disadvantages as infrastructure complexity outpaces human management capacity. The evidence presented here should encourage and guide organizations on this important journey toward intelligent cloud infrastructure management.

REFERENCES

1. T. Khan, W. Tian, G. Zhou, S. Ilager, M. Gong, and R. Buyya, “Machine learning (ML)-centric resource management in cloud computing: A review and future directions,” Journal of Network and Computer Applications, vol. 204, p. 103405, Aug. 2022, doi: 10.1016/j.jnca.2022.103405.
2. F. Ullah, S. Dhingra, X. Xia, and M. A. Babar, “Evaluation of distributed data processing frameworks in hybrid clouds,” Journal of Network and Computer Applications, vol. 224, p. 103837, Apr. 2024, doi: 10.1016/j.jnca.2024.103837.
3. O. Tomarchio, D. Calcaterra, and G. D. Modica, “Cloud resource orchestration in the multi-cloud landscape: a systematic review of existing frameworks,” Journal of Cloud Computing, vol. 9, no. 1, Sep. 2020, doi: 10.1186/s13677-020-00194-7.
4. J. Park, U. Kim, D. Yun, and K. Yeom, “Approach for Selecting and Integrating Cloud Services to Construct Hybrid Cloud,” Journal of Grid Computing, vol. 18, no. 3, pp. 441–469, May 2020, doi: 10.1007/s10723-020-09519-x.
5. Grand View Research, “Global Hybrid Cloud Market Size & Outlook, 2023–2030," 2024. [Online].Available: https://www.grandviewresearch.com/horizon/outlook/hybrid-cloud-market-size/global. Accessed: Jan. 24, 2026
6. A. Berenberg and B. Calder, “Deployment Archetypes for Cloud Applications,” ACM Computing Surveys, vol. 55, no. 3, pp. 1–48, Feb. 2022, doi: 10.1145/3498336.
7. A. Ullah et al., “Orchestration in the Cloud-to-Things compute continuum: taxonomy, survey and future directions,” Journal of Cloud Computing, vol. 12, no. 1, Sep. 2023, doi: 10.1186/s13677-023-00516-5
8. S. Mangalampalli, G. R. Karri, and U. Kose, “Multi objective trust aware task scheduling algorithm in cloud computing using whale optimization,” Journal of King Saud University - Computer and Information Sciences, vol. 35, no. 2, pp. 791–809, Feb. 2023, doi: 10.1016/j.jksuci.2023.01.016.
9. O. Ghandour, S. El Kafhali, and M. Hanini, “Adaptive workload management in cloud computing for service level agreements compliance and resource optimization,” Computers and Electrical Engineering, vol. 120, p. 109712, Dec. 2024, doi: 10.1016/j.compeleceng.2024.109712.
10. P. Y. Cheah, “Data governance, policies and Data Access,” The Data Life Cycle: Practices and Policies, Jul. 2024, doi: 10.48060/tghn.144.
11. I. Behera and S. Sobhanayak, “Task scheduling optimization in heterogeneous cloud computing environments: A hybrid GA-GWO approach,” Journal of Parallel and Distributed Computing, vol. 183, p. 104766, Jan. 2024, doi: 10.1016/j.jpdc.2023.104766.

ISSN: 2178-7727

12. N. Chaurasia, M. Kumar, A. Vidyarthi, K. Pal, and A. Alkhayyat, “An efficient and optimized Markov chain-based prediction for server consolidation in cloud environment,” Computers and Electrical Engineering, vol. 108, p. 108707, May 2023, doi: 10.1016/j.compeleceng.2023.108707.
13. L. Zhu, F. Wu, Y. Hu, K. Huang, and X. Tian, “A heuristic multi-objective task scheduling framework for container-based clouds via actor-critic reinforcement learning,” Neural Computing and Applications, vol. 35, no. 13, pp. 9687–9710, Mar. 2023, doi: 10.1007/s00521-023-08208-6.
14. D. Wu, X. Wang, X. Wang, M. Huang, R. Zeng, and K. Yang, “Multi-objective optimization-based workflow scheduling for applications with data locality and deadline constraints in geo-distributed clouds,” Future Generation Computer Systems, vol. 157, pp. 485–498, Aug. 2024, doi: 10.1016/j.future.2024.04.004.
15. R. Jeyaraj, A. Balasubramaniam, A. K. M.A., N. Guizani, and A. Paul, “Resource Management in Cloud and Cloud-influenced Technologies for Internet of Things Applications,” ACM Computing Surveys, vol. 55, no. 12, pp. 1–37, Mar. 2023, doi: 10.1145/3571729.
16. Z. Ahamed, M. Khemakhem, F. Eassa, F. Alsolami, and A. S. A.-M. Al-Ghamdi, “Technical Study of Deep Learning in Cloud Computing for Accurate Workload Prediction,” Electronics, vol. 12, no. 3, p. 650, Jan. 2023, doi: 10.3390/electronics12030650.
17. I. Pintye, J. Kovács, and R. Lovas, “Enhancing Machine Learning-Based Autoscaling for Cloud Resource Orchestration,” Journal of Grid Computing, vol. 22, no. 4, Oct. 2024, doi: 10.1007/s10723-024-09783-1.
18. G. Zhou, W. Tian, R. Buyya, R. Xue, and L. Song, “Deep reinforcement learning-based methods for resource scheduling in cloud computing: a review and future directions,” Artificial Intelligence Review, vol. 57, no. 5, Apr. 2024, doi: 10.1007/s10462-024-10756-9.
19. P. Singh, V. Prakash, G. Bathla, and R. K. Singh, “QoS aware task consolidation approach for maintaining SLA violations in cloud computing,” Computers and Electrical Engineering, vol. 99, p. 107789, Apr. 2022, doi: 10.1016/j.compeleceng.2022.107789.
20. L. Wang, “Optimizing resource scheduling for cloud workloads running in data centers”, doi: 10.14711/thesis-991012985999103412.
21. V. Malothu, “Optimized Resource Allocation for High-Performance Cloud Systems using Machine Learning,” International Journal of Intelligent Systems and Applications in Engineering, Sep. 2024, doi: 10.17762/ijisae.v12i22s.7975.
22. J. Sun and H. Cho, “A Lightweight Optimal Scheduling Algorithm for Energy-Efficient and Real-Time Cloud Services,” IEEE Access, vol. 10, pp. 5697–5714, 2022, doi: 10.1109/access.2022.3141086.
23. L. Wu, R. Ding, Z. Jia, and X. Li, “Cost-Effective Resource Provisioning for Real-Time Workflow in Cloud,” Complexity, vol. 2020, pp. 1–15, Mar. 2020, doi: 10.1155/2020/1467274.
24. H. Eljak et al., “E-Learning-Based Cloud Computing Environment: A Systematic Review, Challenges, and Opportunities,” IEEE Access, vol. 12, pp. 7329–7355, 2024, doi: 10.1109/access.2023.3339250.
25. A. Asghari, M. K. Sohrabi, and F. Yaghmaee, “Online scheduling of dependent tasks of cloud’s workflows to enhance resource utilization and reduce the makespan using multiple reinforcement learning-based agents,” Soft Computing, vol. 24, no. 21, pp. 16177–16199, Apr. 2020, doi: 10.1007/s00500-020-04931-7.
26. X. Chen, L. Yang, Z. Chen, G. Min, X. Zheng, and C. Rong, “Resource Allocation With Workload-Time Windows for Cloud-Based Software Services: A Deep Reinforcement Learning Approach,” IEEE Transactions on Cloud Computing, vol. 11, no. 2, pp. 1871–1885, Apr. 2023, doi: 10.1109/tcc.2022.3169157.

ISSN: 2178-7727

27. D. Khalasi, D. Bathwar, J. Bhatia, M. Kumhar, and V. Thumar, “Secure and Explainable Artificial Intelligence (XAI) in Cloud Ecosystems: Challenges and Opportunities,” ICT Infrastructure and Computing, pp. 553–567, 2023, doi: 10.1007/978-981-99-4932-8_51.
28. W. Guo, F. Zhuang, X. Zhang, Y. Tong, and J. Dong, “A comprehensive survey of federated transfer learning: challenges, methods and applications,” Frontiers of Computer Science, vol. 18, no. 6, Jul. 2024, doi: 10.1007/s11704-024-40065-x.