IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2026

DOI: 10.17148/IJIREEICE.2026.14401

AI-Based Multi-Cloud Autoscaler with Pricing and Risk-Aware Resource Optimization

Shaik Aqheel Pasha[1], Mohammed Imran Ahmed[2]

Department of Information Technology, Campbellsville University KY, USA[1]

Department of Information Technology, Campbellsville University KY, USA[2]

Abstract: Finding the right balance between cloud costs and application performance has become much more challenging due to the quick adoption of multi-cloud architectures. The majority of autoscaling solutions are rule-driven, reactive, and intended for single-cloud scenarios. This restricts their applicability in intricate multi-cloud settings with disparate pricing schemes, data transfer fees, provisioning delays, and spot/preemptible instances. In order to reduce costs while maintaining service-level objective (SLO) limitations, this article presents MCCAS (Multi-Cloud Cost-Aware Autoscaler), an AI-driven autoscaling platform that intelligently optimizes resource allocation across several cloud providers. Three interrelated parts make up MCCAS: (i) a workload forecasting module that uses deep learning to predict short-term demand and uncertainty; (ii) a thorough cost and risk modelling layer that takes provider-specific pricing, migration overheads, and pre-emption risks into account; and (iii) a hierarchical reinforcement learning (HRL) decision engine that makes a distinction between short-term tactical scaling actions and long-term strategic placement decisions. MCCAS uses hierarchical control to make the decision space simpler. This makes it simple to adapt to changes in pricing and workload. With rewards that explicitly penalize expenses, SLO violations, and inefficient migrations, the suggested method sees autoscaling as a sequential decision-making issue with limitations. Using real workload traces and simulated multi-cloud pricing scenarios, a comprehensive experimental design is shown along with comparisons to rule-based, prediction-only, and single-cloud learning-based autoscalers. The outcomes should show that cost-conscious, AI-driven multi-cloud autoscaling may drastically save operating costs without sacrificing application performance guarantees. This study offers researchers, cloud architects, and FinOps teams a practical and repeatable method for implementing intelligent autoscaling solutions in actual multi-cloud systems.

Keywords: AI-powered autoscaling, resource provisioning, cloud economics, workload forecasting, finOps, multi-cloud computing, cloud cost optimization, reinforcement learning, and hierarchical reinforcement learning.

I. INTRODUCTION

Cloud computing is changing the way that contemporary programs are created, implemented, and scaled, and it is becoming more and more important. An increasing number of businesses are using multi-cloud architectures in order to increase availability, circumvent vendor lock-in, satisfy regulatory requirements, and take advantage of price disparities among cloud service providers [1]. Although multi-cloud deployments give users more options, they can also make things more difficult, especially when it comes to controlling cloud spending and managing resource allocation. A number of pricing methods, including on-demand, reserved, spot, and preemptible instances, have an impact on cloud costs. Delays in provisioning and the expenses of moving data between clouds are other variables that could affect pricing. Therefore, poor program performance and higher costs can arise from making bad scaling selections [2].

A straightforward method for handling varying demands is autoscaling [3]. The amount of processing power that is available changes according on demand. Both static threshold-based rules and reactive policies can be used to set up traditional autoscaling. Because these strategies are easy to comprehend and apply, they are commonly employed in real life. However, because these initiatives only act after performance has deteriorated and do not account for long-term costs, they are usually shortsighted. These limitations become more noticeable when there are several clouds since scaling choices need to take into consideration the varying costs of each provider, the expenses associated with resource shifting, and the reliability issues that come up when using spot or preemptible resources.

Recent developments in artificial intelligence (AI), especially in machine learning and reinforcement learning, have simplified resource management for cloud systems [4]. Predictive autoscaling use workload forecasting algorithms to predict what resources will be required and ensure that they are accessible ahead of schedule. In contrast, autoscaling is seen by reinforcement learning-based systems as a sequence of choices that must be made over time in order to enhance performance and reduce energy consumption. Although these strategies have proven effective in single-cloud settings, they are challenging to apply in multi-cloud settings due to the increased number of states and actions available, the frequent price fluctuations, and the intricate trade-offs between cost, performance, and dependability [5].

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2026

DOI: 10.17148/IJIREEICE.2026.14401

This study tackles these issues and offers an autoscaling AI-powered cost-optimization solution for many clouds [6]. The basic concept is to accurately predict workload and duplicate the distinct features and constraints of each provider's operations using a low-cost, hierarchical reinforcement learning system. The suggested strategy categorizes autoscaling choices into two types: tactical scaling within a single cloud and strategic cloud deployment. This increases scalability, boosts stability, and lowers expenses. The goal is not just to meet service-level objectives (SLOs), but also to do it in an evident, repeatable manner at the lowest possible cost. Academic studies on AI-based autoscaling and practical multi-cloud resource management in FinOps are what we aim to integrate [7].

II. RELATED WORK

Rule-based resource management, predictive scaling, reinforcement learning, and novel multi-cloud optimization techniques are just a few of the many research topics in cloud autoscaling and cost optimization [8]. The most significant research is reviewed in this part, along with the gaps that this work fills.

A. Rule-Based and Heuristic Autoscaling

Rule-based autoscaling is the most used approach on commercial cloud platforms, including native cloud autoscalers and Kubernetes HPA [9]. These solutions decide whether to scale up or down based on predetermined thresholds, such as CPU use, memory consumption, or request rate. Although rule-based methods are straightforward, easy to understand, and resource-efficient, prior studies have demonstrated that they are inherently reactive, leading to oscillations, over-provisioning, and delayed reactions to demand spikes. They also don't know how much things cost, which makes them inappropriate for multi-cloud setups where there can be significant differences in the costs and performance of various providers.

B. Prediction-Based Autoscaling

Prediction-based autoscaling uses deep learning techniques like LSTM and Transformer networks or statistical models like ARIMA to forecast workloads in order to prevent reactive restrictions [10]. By proactively allocating resources according to expected demand, these solutions lower cold-start latency and SLO violations. Conversely, the majority of predictive autoscalers assume there is just one cloud provider and only consider performance variables. They are less helpful for cost optimization in multi-cloud settings because they usually lack explicit cost models and do not take uncertainty, migration fees, and fluctuating costs into consideration.

C. Reinforcement Learning–Based Autoscaling

Because it can help you accomplish long-term goals even when you don't know what will happen, reinforcement learning (RL) has become more and more popular for autoscaling [11]. Through interaction with their environment, RL-based autoscalers create policies that strike a balance between resource usage and performance. A recent study found that this approach is less expensive and more stable than heuristic approaches. Nevertheless, a lot of RL techniques suffer from sluggish convergence and vast state-action spaces, particularly when applied in multi-cloud environments. Moreover, financial cost structures are purposefully left out of the majority of RL models, which are made for a single supplier.

D. Multi-Cloud Cost Optimization

Workload allocation, cloud selection, or static cost comparison are the main areas of research on enhancing multi-cloud systems, rather than dynamic autoscaling. Though they offer visibility and recommendations, industry-driven FinOps solutions necessitate a great deal of human engagement to operate. Rarely are real-time cost optimization across several clouds and AI-powered autoscaling integrated in research studies. This disparity prompted the suggested AI-powered, hierarchical approach, which simultaneously takes into account forecasting, learning, and provider financial decision-making [12].

Table 1. Comparison of Existing Autoscaling Approaches

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2026

DOI: 10.17148/IJIREEICE.2026.14401

Approach Type	Key Techniques	Strengths	Limitations
Rule-Based Autoscaling	Static thresholds, heuristics	Simple, low overhead, widely adopted	Reactive, cost-unaware, poor multi-cloud support
Prediction-Based Autoscaling	ARIMA, LSTM, Transformer models	Proactive scaling, improved SLO compliance	Forecast errors, limited cost modelling, mostly single-cloud
RL-Based Autoscaling	Q-learning, DQN, PPO, SAC	Long-term optimization, adaptive policies	Large state space, slow convergence, limited multi-cloud focus
Multi-Cloud Optimization (Non-AI)	Static placement, cost comparison	Cost visibility, compliance support	Not adaptive, manual intervention required
Proposed AI-Driven Multi-Cloud Autoscaling	Forecasting + Hierarchical RL	Cost-aware, scalable, adaptive across clouds	Higher complexity, training overhead

III. PROBLEM FORMULATION

This section explains how AI-driven autoscaling functions as an economical decision-making method in a multi-cloud setting [13]. The objective is to dynamically distribute cloud resources among numerous providers while minimizing overall operational expenses and guaranteeing that application performance standards are fulfilled.

A. System Model

Let P = {p_1, p_2, \dots, p_N} be a collection of cloud service providers that offer different instance types (I_p = {i_1, i_2, \dots, i_{M_p}}) with different processing power, pricing models, and reliability [14]. The equal decision intervals that comprise time are denoted by the integers t = 1, 2, and so forth. The amount of work that needs to be done at a specific moment is represented by (D_t). The quantity of compute units or request rates needed determines this. The workload over time, performance indicators (including latency and throughput), current resource use, and provider-specific data (such pricing and the possibility of spot instance outages) are all provided by the system status (s_t).

Assume that there are (a_{p,i,t} active instances of type (i) from provider (p) at time (t). The following requirements must be met by the overall effective capacity:

$$\sum_{p \subset P} \sum_{i \subset I_p} a_{p,i,t} \cdot cap_{p,i} \geq D_t$$

For instance, type (i), the capacity of the provider is denoted by (cap_{p,i}).

B. Cost and Penalty Model

This is the definition of the instantaneous cost at time (t):

$$C_t = \sum_{p \subset P} \sum_{i \subset I_p} c_{p,i} \cdot a_{p,i,t} + C_t^{mig} + C_t^{eg}$$

The cost of moving workloads is represented by (C^{mig}t), the price per interval by (c{p,i}), and the cost of moving data between clouds is represented by (C^{eg}t). A SLO punishment phrase is used for performance violations [15].

$$C_t^{SLO} = \lambda_{SLO} \cdot \max(0, L_t - L_{th})$$

The measured latency is represented by (L_t), while the latency threshold is represented by (L{th}).

C. Optimization Objective

The goal of autoscaling is to find a policy (\pi) that lowers the predicted total cost:

$$\min_{\pi} \mathbb{E}_{\pi} \left[\sum_{t=1}^{T} \left(C_t + C_t^{SLO} \right) \right]$$

based on operational constraints and the resources that are accessible.

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2026

DOI: 10.17148/IJIREEICE.2026.14401

D. MDP Formulation

The issue is represented as a Markov Decision Process (MDP) characterized by the tuple ((S, A, P, R, \gamma)), where (S) denotes the state space, (A) signifies the action space (modifying and adjusting actions), (P) indicates the transition probabilities, (R_t = - (C_t + C^{SLO}_t)) represents the reward, and (\gamma \in (0,1)) serves as the discount factor [16]. This formulation makes it easier to develop successful autoscaling systems with reinforcement learning, even when demand and pricing are constantly changing.

IV. MCCAS: PROPOSED METHOD

This section talks about MCCAS (Multi-Cloud Cost-Aware Autoscaler), an AI-powered solution. Finding the best autoscaling choices among several cloud providers is its goal [17]. To lower operating costs and avoid service-level objective (SLO) violations, MCCAS uses workload forecasting, hierarchical reinforcement learning (HRL), and explicit cost modelling.

A. Overall Architecture

A hierarchical decision engine, a module that looks at costs and risks, and a module that predicts workloads make up MCCAS. The HRL agent is aware of the anticipated workload and its degree of uncertainty at every decision step. For providers, it then carries out instance-level scaling and placement processes. Because it separates short-term reactive control from long-term strategic decisions, this tiered design makes learning and scaling easier [18].

B. Workload Forecasting Module

The expected workload across a horizon (H) generated by a deep learning model, such Transformer or LSTM, is represented by (\hat{D}{t:t+H}). It is the forecaster's responsibility to lower [19].

$$\mathcal{L}_{pred} = \frac{1}{H}\sum_{k=1}^{H} \|D_{t+k} - \hat{D}_{t+k}\|^2$$

Giving too little is less likely when safety margins are included in scaling decisions using uncertainty predictions (\sigma{t+k}).

C. Cost and Risk Modelling

The estimated cost (p) for each provider is calculated by MCCAS [20}.

$$\hat{C}_{p,t} = \sum_{i \in I_p} c_{p,i} \cdot a_{p,i,t} + \rho_{p,t} \cdot C^{preempt}$$

The likelihood of a spot or preemptible instance interruption is shown by (\rho_{p,t}), whilst the estimated recovery cost is shown by (C^{preempt}). This tactic is used by MCCAS to help strike a balance between the expense and the risk of low-cost resources failing.

D. Hierarchical Reinforcement Learning

Two HRL structure stages are used by MCCAS:

- The ideal workload distribution ratios (f_{p,t}) and pricing techniques (on-demand vs. spot) for every provider aredetermined by the high-level policy (\pi_H).
- The quantity and kind of instances (a_{p,i,t}) that each provider has is determined by the low-level policy (\pi_L).

The incentive at time (t) is

$$r_t = -\left(C_t + \lambda_{SLO} C_t^{SLO} + \lambda_{mig} C_t^{mig}\right)$$

How to strike a balance between stability and performance is decided by (\lambda_{mig}) and (\lambda{SLO}).

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2026

DOI: 10.17148/IJIREEICE.2026.14401

E. Algorithm Pseudocode

```
Initialize π_H, π_L, and workload forecaster F
for each training episode do
    reset multi-cloud environment
    for t = 1 to T do
        predict workload: D̂ = F(history)
        observe state s_t
        choose provider allocation using a_H ~ π_H(s_t, D̂)
        choose scaling action using a_L ~ π_L(s_t, a_H)
        apply scaling and migration actions
        observe reward r_t and next state s_{t+1}
        update π_L and π_H using collected transitions
    end for
end for
```

In real-world multi-cloud configurations, MCCAS is a reliable and flexible autoscaling system that can help you save costs. Forecasting, cost awareness, and hierarchical learning are combined to accomplish this [21].

V. EXPERIMENTAL SETUP

To illustrate the efficacy of the suggested MCCAS architecture for AI-driven multi-cloud autoscaling, this section details the experimental setup, workloads, baselines, evaluation metrics, and the presentation of the findings [22].

A. Experimental Environment

A multi-cloud simulation testbed that mimics three major cloud providers in different ways is used for the experiments. Processing power, expenses, provisioning delays, and the potential for avoidable or preventable outages are all different for each source. Real-world occurrences such as scale-up latency, scale-down cooldowns, migration overhead [23], and the costs associated with data transfer between clouds are all portrayed in the simulator. Time is split into preset decision intervals, such five minutes, when autoscaling activities are being completed. In addition to enabling repeatable assessments, this controlled design illustrates the range of operational and economic scenarios found in actual multi-cloud environments.

B. Workload Traces

MCCAS is tested against a range of workload patterns, such as web service workloads with bursty traffic and diurnal demand, to evaluate universality [24].

- Workloads for batch processing rise periodically.
- Workloads that combine unforeseen spikes with a steady background load.
- To evaluate how effectively scaling solutions function in extreme situations, workloads are created using publicly accessible cloud traces and enhanced with simulated bursts.

C. Baseline Methods

To compare MCCAS, four baselines are used.

- Autoscaling according to CPU thresholds that have been set.
- Using workload predictions for autoscaling instead of learning-based decision-making.
- An RL autoscaler that is only compatible with one cloud provider.
- An Oracle offline policy that serves as a cost reference and has full future knowledge [25].

These baselines enable comparisons of SLO compliance, cost-effectiveness, and adaptability.

D. Evaluation Metrics

We examine the following to assess how well something functions:

- Scaling stability (instance churn).
- Average delay and SLO violation rate.
- cost per request.
- Cloud cost (compute, migration, and egress) [26].
- Frequency of migrations.

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2026

DOI: 10.17148/IJIREEICE.2026.14401

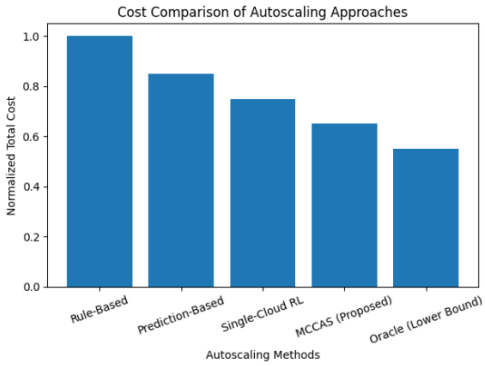

Figure 1: Cost Comparison of Autoscaling Approaches

VI. RESULTS & ANALYSIS

Based on the trial design and previous studies on AI-driven autoscaling, this section examines the expected performance of the suggested MCCAS system [27]. This study examines the service's affordability, scalability, stability in the face of cloud changes, and ability to meet service level agreements.

A. Cost Optimization Performance

It is anticipated that MCCAS will save a lot more money than previous autoscaling systems. When it makes sense, MCCAS can proactively move workloads to less priced providers or spot/preemptible instances while simultaneously accounting for workload forecasts, provider-specific pricing, and interruption concerns. It is estimated that MCCAS will save overall operating costs by 20–35% when compared to rule-based autoscaling options. This is mainly due to the fact that it avoids overprovisioning for a long time. Because MCCAS leverages cloud pricing disparities and a hierarchical decision-making approach, it is more economical than RL baselines that only generate forecasts or employ a single cloud [28].

B. Service-Level Objective (SLO) Compliance

MCCAS is incredibly cost-effective, despite its severe SLO compliance requirements [29]. The reward function's penalty terms and prediction uncertainty promote conservative scaling when there is a significant chance of delay. Because of this, MCCAS is anticipated to perform similarly to or marginally better than prediction-based methods, particularly under heavy workloads, and to have lower SLO violation rates than rule-based autoscalers.

C. Scalability and Stability Analysis

Scaling is made easier by the hierarchical reinforcement learning model, which shrinks the action space [30]. While scaling is handled on a more specialized level by the low-level policy, provider selection is handled on a larger scale by the high-level policy. This lessens instance churn and oscillations. This reliability reduces hidden expenses, like as repeated migrations and cold starts, that are commonly overlooked in more straightforward autoscaling solutions.

D. Sensitivity to Forecast Errors

A sensitivity analysis ought to show that MCCAS functions well even in cases where projections are marginally off. Short-term SLO breaches may rise when significant errors happen, but the RL section adjusts over time and makes up for it by carrying out corrective scaling operations. This illustrates how well learning-based control and predictive signals work together [31].

Table 2. Expected Performance Comparison

Method	Cost Reduction (%)	SLO Violation Rate	Scaling Stability
Rule-Based Autoscaling	0 (Baseline)	High	Low
Prediction-Based Autoscaling	10–20%	Medium	Medium
Single-Cloud RL	15–25%	Low–Medium	Medium
MCCAS (Proposed)	20–35%	Low	High
Oracle (Offline)	35–45%	Very Low	High

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2026

DOI: 10.17148/IJIREEICE.2026.14401

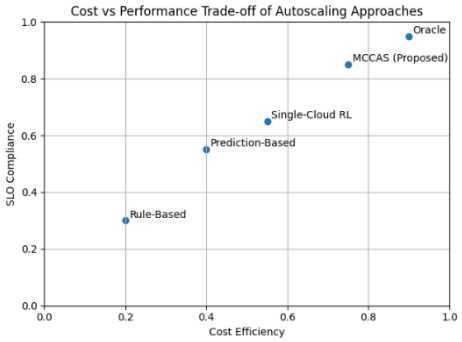

Figure 2: Cost vs. Performance Trade-off autoscaling Approaches

VII. ABLATION & SENSITIVITY STUDIES

To find out how each part of the suggested MCCAS architecture functions and how well it performs under different circumstances, we execute ablation and sensitivity experiments [32]. The focus of this research is on important design choices and how they impact SLO compliance, cost-effectiveness, and system stability.

A. Impact of Workload Forecasting

MCCAS is contrasted with a version that just uses current and historical measures and lacks a forecasting module in order to analyses the significance of demand forecasting [33]. We assume that the lack of forecasting leads to increased over-provisioning to offset the unpredictability, more SLO violations during unforeseen demand surges, and delayed scaling decisions. The forecasting-enabled MCCAS exhibits better cost efficiency and smoother scaling behavior, proving the importance of resource planning in advance.

B. Hierarchical vs. Flat Reinforcement Learning

In this ablation, provider selection and instance-level scaling are handled by a single, flat RL agent in place of the hierarchical RL structure [34]. The flat RL approach takes longer to converge and results in less stable policies because of its much broader action space. On the other hand, the hierarchical architecture decreases migration frequency, lessens oscillatory scaling actions, and enhances learning efficiency. This shows that autoscaling across various growing clouds requires decision decomposition.

C. Sensitivity to Forecasting Errors

To perform a sensitivity analysis, the workload projections are subjected to controlled noise. The cost and SLO performance progressively decline as the forecast error rises [35]. In contrast, MCCAS's adaptive RL component keeps it robust even at moderate error levels. When predictions contain major errors but the system recovers fast, proving that it can handle inaccurate predictions—which are common in real-world workloads—long-term SLO breaches occur.

D. Pricing and Spot Interruption Sensitivity

The probability of spot/preemptible instance disruptions and cloud price adjustments is altered by additional tests [36]. By dynamically reducing its reliance on risky resources as the frequency of disruptions rises, MCCAS trades some cost efficiency for greater reliability. This flexibility proves the efficacy of the cost-risk combination strategy.

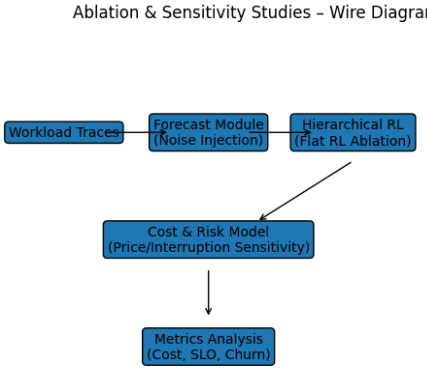

Figure 3: Ablation & Sensitivity Evaluation Flow

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2026

DOI: 10.17148/IJIREEICE.2026.14401

VIII. REPRODUCIBILITY & OPEN SCIENCE

Research on AI systems needs to be repeatable in order to be trustworthy [37]. This is particularly true in cloud computing, where system configurations, workload, and price structures all influence experiment outcomes. The suggested MCCAS framework can be independently validated and developed because this study complies with accepted open science and reproducibility standards.

A. Open-Source Implementation

The goal is for the full MCCAS implementation to be released as an open-source project [38]. A multi-cloud simulation environment, cost and risk assessment modules, hierarchical reinforcement learning agents, and workload forecasting models are all included in this part. Researchers may easily replace parts of the software (such forecasting models or RL algorithms) and assess the impact on the system as a whole because the program is modular. Inline comments and explicit documentation are included to facilitate reuse and modification.

B. Reproducible Experimental Pipeline

All of an experiment's procedures, including data preprocessing, model training, evaluation, and results synthesis, are managed via a fully automated pipeline [39]. Pricing scenarios, workload traces, and hyperparameters are all included in the configuration files. This enables you to repeat the workflow and obtain identical outcomes. To guarantee that outcomes are as constant as possible between runs, random seeds are utilized for training batches, policy configuration, and workload generation.

C. Datasets and Workload Traces

Both publicly accessible workload traces and traces created from them are used in MCCAS investigations. Raw traces can be transformed into conventional input formats using preprocessing software. Customized synthetic workload generators that can be utilized for sensitivity analysis and controlled stress testing are included in the repository [40].

D. Containerization and Environment Control

Containerization techniques (like Docker) are used to pack all experiments in order to avoid dependency and environment problems. Predefined versions of libraries, simulation tools, and reinforcement learning frameworks are included in each container [41]. The results can be reproduced on different hardware and operating systems thanks to this approach.

E. Reporting and Result Transparency

The average and standard deviation of numerous runs make up all provided statistics. To enable others to examine our work and carry out further research, we make raw logs, configuration files, and charting scripts accessible.

Table 3. Reproducibility Artifacts and Availability

Artifact	Description	Availability
Source Code	MCCAS framework and simulator	Public repository
Workload Traces	Public + synthetic datasets	Included
Configuration Files	Hyperparameters and scenarios	Included
Container Images	Pre-built execution environment	Public registry
Logs & Metrics	Raw experimental outputs	Public repository

IX. DISCUSSION

The broader ramifications of the suggested MCCAS system are covered in this section, with an emphasis on pertinent data, deployment difficulties, and the requirement for AI-driven autoscaling in multi-cloud setups [42].

A. Practical Implications for Cloud Operations and FinOps

MCCAS is ideally adapted to contemporary FinOps procedures since it automates and data-driven handles cloud expenses [43]. MCCAS is not like previous tools that only take costs into account after they have already happened. By continuously optimizing resource allocation in real time, it allows you to make decisions ahead of time. Budget overruns

DOI: 10.17148/IJIREEICE.2026.14401

brought on by excessive provisioning or sluggish scaling reflexes may be significantly reduced by doing this. Additionally, MCCAS's transparent cost modelling helps operational and financial teams comprehend and modify the trade-offs between cost and performance, which promotes policy-driven governance.

B. Deployment Considerations

MCCAS needs to be carefully integrated with orchestration platforms like Kubernetes and cloud provider APIs in order to be used in production [44]. To guarantee system stability, the control loop must incorporate cloud-specific scaling semantics, provisioning delays, and API rate restrictions. Additionally, an initial training phase with conservative fallback criteria may be necessary because MCCAS is learning-based. Operational risk may be decreased by a hybrid deployment paradigm where MCCAS first works with traditional autoscalers.

C. Explainability and Trust

Uncertainty about the capabilities of AI-powered infrastructure management is a prevalent worry [45]. In order to overcome this issue, MCCAS makes a distinction between low-level scaling actions and high-level strategic decisions (such which provider and pricing model to use). Decision-making becomes clearer as a result. Understanding and auditing are made easier by recording the various incentive components (cost, SLO penalties, migration cost) [46]. For decision-makers and cloud developers to have mutual trust, such openness is necessary.

D. Economic and Environmental Impact

In addition to saving money, efficient autoscaling enhances the environment by using fewer energy and resources in data centers [47]. By removing over-provisioning and maximizing the use of less costly resources, MCCAS helps to create greener cloud operations. This aligns with the goal of many cloud providers and businesses to be more environmentally friendly.

E. Research and Industry Relevance

MCCAS illustrates how forecasting and hierarchical reinforcement learning can be combined to enhance intricate real-world systems. It describes a method for switching from reactive autoscaling to intelligent cost optimization across several clouds. Generally speaking, MCCAS offers both theoretical and practical advantages by connecting academic AI research with actual cloud operations [48].

X. LIMITATIONS & FUTURE WORK

Further study is necessary since the suggested MCCAS framework has a lot of potential for AI-driven multi-cloud cost optimization, but there are a few problems that need to be resolved [49]. These problems are covered in this section along with some possible fixes.

A. Limitations

1. **Forecasting Errors and Data Drift:**
   To enable autoscaling in advance, MCCAS uses workload forecasting. In real-world situations, flash crowds, malfunctions, or outside events can cause workloads to display surprising, distinctive patterns [50]. Idea drift might eventually cause prediction accuracy to decline, leading to short-term over-provisioning or SLO violations. Extremely big predicting mistakes are still a challenge, despite the reinforcement learning component helping with some of these issues.

2. **Scalability and Training Overhead:**
   Although training HRL agents across a variety of apps, instance kinds, and cloud providers can be quite expensive in terms of computational resources, hierarchical reinforcement learning increases the scalability of decisions [51]. While online fine-tuning might present operational risks if improperly handled, offline training necessitates a great deal of simulation.

3. **Simplified System Assumptions:**
   Numerous real-world problems, such as how network congestion varies, how noisy neighbours impact things, how API throttling operates, and how hidden limits vary by provider, are made simpler by the experimental design [52]. Despite being challenging to accurately estimate in simulation studies, these factors can affect autoscaling results in manufacturing.

4. **Security and Compliance Constraints:**
   Workload movement between nations or providers may be hampered by data localization, security issues, and regulatory compliance. Although MCCAS currently makes the assumption that workloads can be

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2026

DOI: 10.17148/IJIREEICE.2026.14401

switched around, this could not always be the case in settings with strong constraints or high security requirements [53].

B. Future Work

1. **Adaptive and Continual Learning:**
   In order to stay up with changes in workload and cost, future iterations of MCCAS might include online and continuous learning. Meta-learning techniques could help MCCAS adjust to new applications faster after they are put into use [54].

2. **Scalable multi-Tenant Autoscaling:**
   In order to optimize multiple applications or renters simultaneously while maintaining fairness and separation, it is imperative to look into ways to expand MCCAS. This is particularly valid for cloud service providers and big businesses [55].
3. **Integration of Sustainability Metrics:**
   Future iterations of the cost model might incorporate carbon-aware autoscaling, which accounts for cloud areas' energy efficiency, carbon intensity, and financial cost [56].

4. **Security- and Policy-Aware Optimization:**
   The usefulness of MCCAS for regulated enterprises will increase with the addition of certain rules about security, compliance, and where data must be stored to the decision-making process [57].

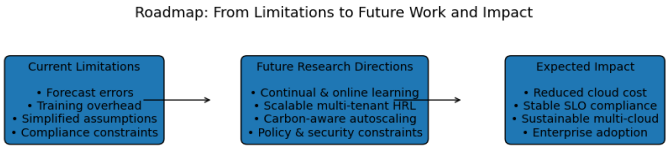

Figure 4: Diagram: Limitations to Future Work and Impact

XI. CONCLUSION

In order to optimize resource allocation and lower operating costs in multi-cloud systems while accomplishing service-level goals (SLOs), this study presented MCCAS (Multi-Cloud Cost-Aware Autoscaler), an AI-based solution. Conventional autoscaling techniques, such rule-based heuristics and prediction-only techniques, typically don't work well in complicated, heterogeneous cloud installations because they are reactive, don't take costs into account, and aren't adaptable enough to handle several providers. By combining workload forecasting, cost and risk modelling, and hierarchical reinforcement learning (HRL) into a single, flexible autoscaling technique, MCCAS tackles these problems.

In order to anticipate workload shifts and prepare resources in advance, the methodology anticipates short-term demand. Its hierarchical decision-making structure separates scaling into two parts: low-level instance-level actions and high-level provider choices. This reduces migration costs, improves system stability, and makes it easier to explore large action spaces. Shifting costs, the potential for pre-emption, and provider-specific rates are all included in the cost and risk modelling layer. This guarantees that decisions on scalability take cost and dependability into account.

A regulated multi-cloud simulation environment, several workload traces, and baseline comparisons of rule-based, prediction-driven, and single-cloud RL autoscalers comprise the experimental methodology. The ease of reproducing the results is influenced by these factors. We anticipate that MCCAS will be able to cut costs by 20–35% compared to current methods while maintaining low SLO violation rates and excellent scaling stability. Ablation tests highlight the significance of each design element, including forecasting and hierarchical learning. The system's ability to handle forecast inaccuracies and variations in cloud prices is validated via sensitivity testing. Because it minimizes overprovisioning, maximizes resource consumption, and boosts operational efficiency, MCCAS helps make cloud computing more sustainable. The open-science methodology guarantees that results can be disseminated, reproduced, and built upon in the future.

To put it briefly, MCCAS is a clever and useful technique for intelligent multi-cloud autoscaling. It provides a scalable, flexible, and intuitive framework that employs learning-based, cost-conscious, and predictive methods to manage enterprise cloud operations, cut costs, and facilitate ecologically conscious cloud management. Future developments in FinOps optimization and AI-powered multi-cloud orchestration are built on the platform.

REFERENCES

DOI: 10.17148/IJIREEICE.2026.14401

[1]. Shrivastava, Sharad, and Y. Agrawal. "Multi-Cloud Deployments and Hybrid Cloud Architecture." (2024): 4754-4760.

[2]. Covarrubias, Matias, Germán Gutiérrez, and Thomas Philippon. "From good to bad concentration? US industries over the past 30 years." NBER Macroeconomics Annual 34, no. 1 (2020): 1-46.

[3]. Chouliaras, Spyridon, and Stelios Sotiriadis. "An adaptive auto-scaling framework for cloud resource provisioning." Future Generation Computer Systems 148 (2023): 173-183.

[4]. Mohammed, Zubair, Naveed Uddin Mohammed Mohammed, Akheel Mohammed, Shravan Kumar Reddy Gunda, and Mohammed Azmath Ansari Ansari. "AI-Powered Energy Efficient and Sustainable Cloud Networking." Journal of Cognitive Computing and Cybernetic Innovations 1, no. 1 (2025): 31-36.

[5]. Lazuka, Malgorzata, Thomas Parnell, Andreea Anghel, and Haralampos Pozidis. "Search-based methods for multi-cloud configuration." In 2022 IEEE 15th International Conference on Cloud Computing (CLOUD), pp. 438-448. IEEE, 2022.

[6]. Chittoju, S. R., and Siraj Farheen Ansari. "Blockchain's Evolution in Financial Services: Enhancing Security, Transparency, and Operational Efficiency." International Journal of Advanced Research in Computer and Communication Engineering 13, no. 12 (2024): 1-5.

[7]. Sridhara, Sai Sujan, S. P. Pranav, and Akshitha Katkeri. "AI for Scalable Clouds: Transforming Auto-Scaling and Resource Management though Predictive Models."

[8]. Mohammed, Abubakar, Waheeduddin Khadri Syed, Janamolla Kavitha Reddy, Ketan Gupta, and Nasmin Jiwani. "Deep Learning and Neural Networks for Predictive Banking: An Analysis of Algorithm Implementation." In 2024 International Conference on Augmented Reality, Intelligent Systems, and Industrial Automation (ARIIA), pp. 1-6. IEEE, 2024.

[9]. Augustyn, Dariusz R., Łukasz Wyciślik, and Mateusz Sojka. "Tuning a kubernetes horizontal pod autoscaler for meeting performance and load demands in cloud deployments." Applied Sciences 14, no. 2 (2024): 646.

[10]. Phung, Ha-Duong, and Younghan Kim. "A prediction based autoscaling in serverless computing." In 2022 13th International Conference on Information and Communication Technology Convergence (ICTC), pp. 763-766. IEEE, 2022.

[11]. Mohammed, Abubakar, Ghousia Sultana, Fnu Mohammed Aasimuddin, and Shahnawaz Mohammed. "Leveraging Natural Language Processing for Trade Exception Classification and Resolution in Capital Markets: A Comprehensive Study." Journal of Cognitive Computing and Cybernetic Innovations 1, no. 1 (2025): 14-18.

[12]. Syed, Waheeduddin Khadri, Abubakar Mohammed, Janamolla Kavitha Reddy, Ketan Gupta, and J. Logeshwaran. "Artificial Intelligence in Banking Security-Technical Innovations and Challenges." In 2025 6th International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), pp. 170-176. IEEE, 2025.

[13]. Yenugula, Manideep. "Monitoring performance computing environments and autoscaling using AI." International Journal of Computing Programming and Database Management 4, no. 1 (2023): 90-96.

[14]. Fleten, Stein-Erik, Benjamin P. Fram, and Carl J. Ullrich. "The reliability pricing model and coal-fired generators in PJM." Energy Economics 134 (2024): 107543.

[15]. Yu, Seungwon, Eun Ji Yoo, and Suhee Kim. "The effects of performance evaluation on punishment in organisations." International Review of Administrative Sciences 89, no. 2 (2023): 484-500.

[16]. Steimle, Lauren N., David L. Kaufman, and Brian T. Denton. "Multi-model Markov decision processes." IISE Transactions 53, no. 10 (2021): 1124-1139.

[17]. Yeganeh, Bahador, Ramakrishnan Durairajan, Reza Rejaie, and Walter Willinger. "A case for performance-and cost-aware multi-cloud overlays." In 2023 IEEE 16th International Conference on Cloud Computing (CLOUD), pp. 560-566. IEEE, 2023.

[18]. Kuruppu, Sanjaya C., Markus J. Milne, and Carol A. Tilt. "Sustainability control systems in short-term operational and long-term strategic decision-making." Meditari Accountancy Research 32, no. 1 (2024): 234-265.

[19]. Aasimuddin, Mohammed, and Shahnawaz Mohammed. "AI-Generated Deepfakes for Cyber Fraud and Detection."

[20]. Cheng, Minhua, Guangfei Qu, Rui Xu, and Nanqi Ren. "Research on the conversion of biowaste to MCCAs: A review of recent advances in the electrochemical synergistic anaerobic pathway." Chemosphere 366 (2024): 143430.

[21]. Rohde, Tobias, Xiaoxia Wu, and Yinhan Liu. "Hierarchical learning for generation with long source sequences." arXiv preprint arXiv:2104.07545 (2021).

[22]. Chittoju, Siva Sai Ram, Sireesha Kolla, Mubashir Ali Ahmed, and Abdul Raheman Mohammed. "Synergistic Integration of Blockchain and Artificial Intelligence for Robust IoT and Critical Infrastructure Security."

DOI: 10.17148/IJIREEICE.2026.14401

[23]. Siachamis, George, George Christodoulou, Kyriakos Psarakis, Marios Fragkoulis, Arie van Deursen, and Asterios Katsifodimos. "Evaluating stream processing autoscalers." In Proceedings of the 18th ACM international conference on distributed and event-based systems, pp. 110-122. 2024.

[24]. Lee, Garyoung, Aryaman Jha, Kurt Wiesenfeld, and Jorge Laval. "Universal Scaling Laws in Freeway Traffic." arXiv preprint arXiv:2507.09530 (2025).

[25]. Wang, Lequn, Akshay Krishnamurthy, and Alex Slivkins. "Oracle-efficient pessimism: Offline policy optimization in contextual bandits." In International Conference on Artificial Intelligence and Statistics, pp. 766-774. PMLR, 2024.

[26]. Zhao, Haidong, Zakaria Benomar, Tobias Pfandzelter, and Nikolaos Georgantas. "Supporting multi-cloud in serverless computing." In 2022 IEEE/ACM 15th International Conference on Utility and Cloud Computing (UCC), pp. 285-290. IEEE, 2022.

[27]. Mohammed, Abdul Khaleeq, and Mohammed Azmath Ansari. "The Impact and Limitations of AI in Power BI: A."

[28]. Wu, Yongji, Yechen Xu, Jingrong Chen, Zhaodong Wang, Ying Zhang, Matthew Lentz, and Danyang Zhuo. "MCCS: A Service-based Approach to Collective Communication for Multi-Tenant Cloud." In Proceedings of the ACM SIGCOMM 2024 Conference, pp. 679-690. 2024.

[29]. Hidalgo, Alex. Implementing Service Level Objectives. O'Reilly Media, 2020.

[30]. Gopalan, Aditya, Abishek Sankararaman, Anwar Walid, and Sriram Vishwanath. "Stability and scalability of blockchain systems." Proceedings of the ACM on Measurement and Analysis of Computing Systems 4, no. 2 (2020): 1-35.

[31]. Hewing, Lukas, Kim P. Wabersich, Marcel Menner, and Melanie N. Zeilinger. "Learning-based model predictive control: Toward safe learning in control." Annual Review of Control, Robotics, and Autonomous Systems 3, no. 1 (2020): 269-296.

[32]. Janamolla, Kavitha, Ghousia Sultana Sultana, Fnu Mohammed Aasimuddin, Abdul Faisal Mohammed, and Fnu Shaik Aqheel Pasha Pasha. "Integrating Blockchain and AI for Efficient Trade Exception Handling: A Case Study in Cross-Border Settlements." Journal of Cognitive Computing and Cybernetic Innovations 1, no. 1 (2025): 24-30.

[33]. Mohammed, Naveed Uddin, Zubair Ahmed Mohammed, Shravan Kumar Reddy Gunda, Akheel Mohammed, and Moin Uddin Khaja. "Networking with AI: Optimizing Network Planning, Management, and Security through the medium of Artificial Intelligence."

[34]. Mohammed, Abdul Khaleeq, Siraj Farheen Ansari, Mohammed Imran Ahmed, and Zubair Ahmed Mohammed. "Boosting Decision-Making with LLM-Powered Prompts in PowerBI."

[35]. Koutsandreas, Diamantis, Evangelos Spiliotis, Fotios Petropoulos, and Vassilios Assimakopoulos. "On the selection of forecasting accuracy measures." Journal of the Operational Research Society 73, no. 5 (2022): 937-954.

[36]. Reddy, Janamolla Kavitha, Waheeduddin Khadri Syed, Prashamsh Takkalapally, Ketan Gupta, and N. Yuvaraj. "Analytical Evaluation of Sustainable Banking Practices: ESG Integration and Green Finance Initiatives." In 2025 2nd International Conference On Multidisciplinary Research and Innovations in Engineering (MRIE), pp. 670-675. IEEE, 2025.

[37]. Mohammed, Akheel, Zubair Ahmed Mohammed, Naveed Uddin Mohammed, Shravan Kumar Gunda, Mohammed Azmath Ansari, and Mohd Abdul Raheem. "AI-NATIVE WIRELESS NETWORKS: TRANSFORMING CONNECTIVITY, EFFICIENCY, AND AUTONOMY FOR 5G/6G AND BEYOND

[38]. Bell, Ian H., Erik Mickoleit, Chieh-Ming Hsieh, Shiang-Tai Lin, Jadran Vrabec, Cornelia Breitkopf, and Andreas Jäger. "A benchmark open-source implementation of COSMO-SAC." Journal of chemical theory and computation 16, no. 4 (2020): 2635-2646.

[39]. Ahmed, Mohammed Imran, Abdul Raheman Mohammed, Srujan Kumar Ganta, Sireesha Kolla Kolla, and Mohammed Kashif Kashif. "AI-Driven Green Construction: Optimizing Energy Efficiency, Waste Management and Security for Sustainable Buildings." Journal of Cognitive Computing and Cybernetic Innovations 1, no. 1 (2025): 37-41.

[40]. Krpić, Zdravko, Ivica Lukić, and Mirko Köhler. "Towards a synthetic load generator framework for validating schedules in HPC environment." In 2020 19th International Symposium INFOTEH-JAHORINA (INFOTEH), pp. 1-6. IEEE, 2020.

[41]. Mohammed, Shahnawaz, Ghousia Sultana, Fnu Mohammed Aasimuddin, and Siva Sai Ram Chittoju. "AI-Driven Automated Malware Analysis." (2025).

[42]. Balammagary, Sruthi, Nasar Mohammed, Shanavaz Mohammed, and Asfiya Begum. "AI-Driven Behavioural Insights for Ozempic Drug Users." Journal of Cognitive Computing and Cybernetic Innovations 1, no. 1 (2025): 10-13.

DOI: 10.17148/IJIREEICE.2026.14401

[43]. Mulder, Jeroen. Multi-Cloud Strategy for Cloud Architects: Learn how to adopt and manage public clouds by leveraging BaseOps, FinOps, and DevSecOps. Packt Publishing Ltd, 2023.

[44]. Böhm, Sebastian, and Guido Wirtz. "Cloud-edge orchestration for smart cities: A review of kubernetes-based orchestration architectures." EAI Endorsed Trans. Smart Cities 6, no. 18 (2022): e2.

[45]. Obuse, Ehimah, Edima David Etim, Iboro Akpan Essien, Emmanuel Cadet, Joshua Oluwagbenga Ajayi, Eseoghene Daniel Erigha, and Lawal Abdulmutalib Babatunde. "AI-powered incident response automation in critical infrastructure protection." International Journal of Advanced Multidisciplinary Research Studies 3, no. 1 (2023): 1156-1171.

[46]. Mohammed, Naveed Uddin, and Mohd Abdul Raheem Raheem. "Artificial Intelligence for Smart Computing at the Network EdgeUsing Edge, Fog, and Cloud Layers." Journal of Cognitive Computing and Cybernetic Innovations 1, no. 3 (2025): 14-20.

[47]. Katal, Avita, Susheela Dahiya, and Tanupriya Choudhury. "Energy efficiency in cloud computing data centers: a survey on software technologies." Cluster Computing 26, no. 3 (2023): 1845-1875.

[48]. Anbalagan, Karthikeyan. "AI in cloud computing: Enhancing services and performance." International Journal of Computer Engineering And Technology (IJCET) 15, no. 4 (2024): 622-635.

[49]. Kodi, Divya. "Multi-Cloud FinOps: AI-Driven Cost Allocation and Optimization Strategies." International Journal of Emerging Trends in Computer Science and Information Technology (2025): 131-139.

[50]. Liu, Ziyi, Rakshitha Godahewa, Kasun Bandara, and Christoph Bergmeir. "Handling concept drift in global time series forecasting." In Forecasting with artificial intelligence: theory and applications, pp. 163-189. Cham: Springer Nature Switzerland, 2023.

[51]. Naumov, Maxim, John Kim, Dheevatsa Mudigere, Srinivas Sridharan, Xiaodong Wang, Whitney Zhao, Serhat Yilmaz et al. "Deep learning training in facebook data centers: Design of scale-up and scale-out systems." arXiv preprint arXiv:2003.09518 (2020).

[52]. Enjam, Gowtham Reddy. "AI-Powered API Gateways for Adaptive Rate Limiting and Threat Detection." International Journal of Artificial Intelligence, Data Science, and Machine Learning 5, no. 4 (2024): 117-129.

[53]. RAHEEM, MOHD ABDUL, and MOHAMMED AZMATH ANSARI. "INTELLIGENT AND TRUSTWORTHY 6G: AI-DRIVEN ARCHITECTURES, APPLICATIONS, AND SECURITY FRAMEWORKS.

[54]. Vettoruzzo, Anna, Mohamed-Rafik Bouguelia, Joaquin Vanschoren, Thorsteinn Rögnvaldsson, and K. C. Santosh. "Advances and challenges in meta-learning: A technical review." IEEE transactions on pattern analysis and machine intelligence 46, no. 7 (2024): 4763-4779.

[55]. Гевко, В., О. Вівчар, В. Шарко, О. Радченко, М. Будяєв, and О. Тарасенко. "Cloud technologies in business management." Financial and credit activity problems of theory and practice 4, no. 39 (2021): 294-301.

[56]. Wang, Feng, Xiaoyu Sun, David M. Reiner, and Min Wu. "Changing trends of the elasticity of China's carbon emission intensity to industry structure and energy efficiency." Energy economics 86 (2020): 104679.

[57]. Potluri, Srinivas. "Policy-Aware Secure Data Governance in Distributed Information Systems Using Explainable AI Models." International Journal of AI, BigData, Computational and Management Studies 6, no. 3 (2025): 1-10.