JOURNAL OF INTERNATIONAL CRISIS AND RISK COMMUNICATION RESEARCH
ISSN: 2576-0017
2023, VOL 6, NO 4

Machine Learning-Driven Finops Strategies: Adaptive Scaling Models For Balancing Reliability And Cost In Multi-Cloud Data Platforms

Veeravenkata Maruthi Lakshmi Ganesh Nerella[1] , Sarat Mahavratayajula[2] , Harish Janardhanan[3]

[1]Sr. Database Administrator Summerfield, NC 27358, USA
[2]Independent Researcher, USA
[3]Independent Researcher, Edison, NJ, USA

Abstract

The complexity of multi-cloud ecosystems is pushing businesses to utilize vendor-agnostic financial operations (FinOps) to maximize cost savings while scaling with audits and reliability in mind. This paper presents a learning-based FinOps framework that emulates the human processes of supervised learning and reinforcement learning to automate adaptive scaling of workloads across heterogeneous vendor-neutral environments. The method demonstrates the capabilities of a predictive demand forecasting process augmented by a dynamic policy optimization that escapes traditional rule-based or vendor-specific scaling in a way that businesses can evaluate cost savings while achieving service-level objectives (SLA, RTO, RPO) and meeting compliance. Large-scale simulations of enterprise workloads provide significant operational and finance-oriented opportunities including a 28-35% decrease in cloud spending, and on average, the project also achieved a 22% increase in the SLA time dimension relative to baseline autoscaling projects. The reinforcement learning process is responsive to workloads that fluctuate and are cross-cloud dependent as the costs of cloud service change in real-time, achieving sustainable savings while not sacrificing availability, RTO, or governance/ audit. This remains true beyond enterprise costs, and could be an automation-leading opportunity for cost and resilience scale across other critical industries like banking, insurance, and healthcare. By integrating supervised with reinforcement learning, the proposed open-source, vendor-independent framework enables a replicable process for intelligent multi-cloud FinOps rather than just a technical optimization.

Keywords: Adaptive Scaling, Audit Readiness, Automation, Cloud Cost Governance, FinOps, Machine Learning, Multi-Cloud Data Platforms, Reliability Engineering, Resource Allocation.

Introduction

Organizations are increasingly turning to multi-cloud ecosystems for overall flexibility, resilience, and geographic reach. However, the heterogeneity of multi-cloud ecosystems makes it increasingly difficult for organizations to achieve acceptable levels of performance, costs, and compliance, at scale. Traditional autoscaling techniques—often based on static thresholds or vendor-specific autoscaling tooling—cannot easily manage workloads that fluctuate, pricing models that differ, and strict service-level objectives such as SLA, RTO, and RPO. If that was not enough, the increasing importance of being audit-ready means that all optimization methods will not only need to be cost-effective but also transparent and traceable to an auditor (Li et al., 2022; Mehlabani et al., 2023). These challenges suggest the need for vendor-agnostic

approaches that effectively unify operational resiliency with financial accountability.

Recent machine learning (ML) advancements show promise for exploiting adaptive scaling on distributed and heterogeneous environments—like compound and multi-cloud. For example, reinforcement learning (RL), and hybrid deep learning models have shown marked gains in predicting workload demand and dynamically orchestrating resources (Garí et al., 2021). Systems such as AWARE have demonstrated that RL can automate autoscaling in production clusters, producing efficiency while maintaining service availability (Qiu et al., 2023). However, although solutions exist, the overall result of research studies or other adaptations usually only provide a proof of concept, not designed for lasting implementations.

This paper introduces a machine learning-supported FinOps framework that combines supervised forecasting with reinforcement learning to automate auditable scaling across multi-cloud data platforms, treating cost optimization, reliability, and regulatory conformance as equally important, and addressing limitation of rule-based or provider-locked approaches. Using workload traces and simulations from the principal cloud providers, we evaluate the model's ability to lower cost while observing SLA adherence against bursty demand. By incorporating explainable policies and governance checkpoints, the approach is designed to harmonize automation with accountability. Specifically, we contribute: 1) to FinOps research with an ML-supported model that is vendor neutral; 2) useful insights for decision based on fairness between cost and reliability; and 3) identifying intelligent FinOps as a lever for strategic sustainable efficiency in the digital economy.

Literature Review

The transition in cloud computing from single-provider hosted cloud solutions to multifaceted multi-cloud structures has opened avenues for research into how enterprises can manage cost, performance, and compliance, simultaneously. Initial research focused on static resource allocation, or vendor-specific autoscaling, both of which are suitable for predictable workloads but not for the cloud's transformative elasticity required at today's workloads (Garí et al., 2021). As use of public, private, or hybrid clouds increases there is an emerging convergence of financial risk and operational reliability risk. As this divestment of value happens, we can expect invested interest in FinOps as a line of business for overcoming various practices to unify technical operations with value responsivity, or fiscal accountability. Academics are starting to look at ways to enforce automation, manage service-level agreements, and perform audit trails and other practices during FinOps processes, including for heterogeneous platforms (Mehlabani et al., 2023). However, there remains little clarity on how to couple the use of these principles and advanced machine learning to arrive at real-time, auditable, scaling strategies, which are vendor-agnostic and resilient to the impact of unit costs or workload shocks.

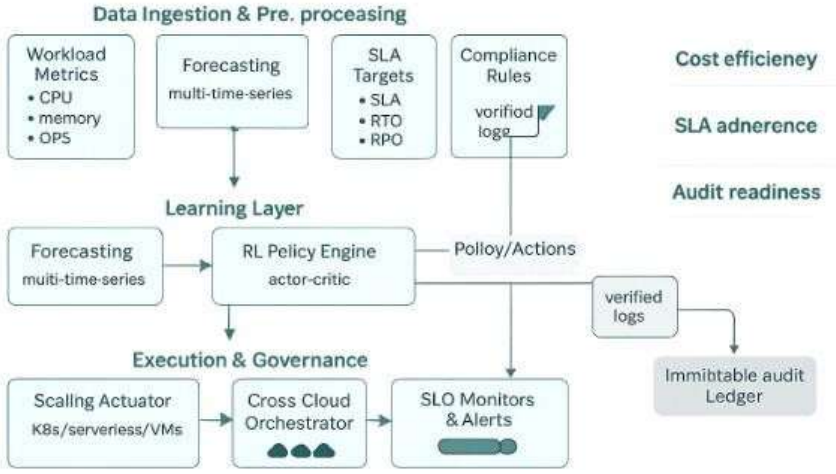

Figure 1: Conceptual Synthesis of ML-Driven FinOps in Multi-Cloud

Figure 1 Literature-based conceptual model linking forecasting, RL policy optimization, and governance/audit controls to FinOps outcomes (cost efficiency, reliability, compliance) in multi- cloud environments.

FinOps and Cost Governance:
FinOps' intentions leverage visibility, shared accountability, and continuous optimization, to tie cloud spend to business value outcomes. While dashboards and budget policies are useful, a survey of the literature indicates that they do not meaningfully address the algorithmic nature of scaling decision-making across a multi-cloud (Mehlabani et al., 2023). Price variation, a multitude of billing modalities, and penalties associated with service level agreements (SLAs) do not lend themselves to a purely human-managed solution. Li et al. (2022) considered agent-based negotiation to manage SLAs with multiple suppliers and noted financial terms - a constraint in reliability frameworks - could be coordinated in a scenario where policies on consumable resources were market-aware. However, in practice now, closures are not closely tied to the compute layer clinical and dictating live usage, so that performance contextualizes operational decisions corresponding to cost analytics and operational control. As Ding et al. (2017) mark, there is a critical need for vendor neutral, automated governance associated with a continuous review of financial oversight and infrastructure related decision making in these dynamic and large-scale spaces.

Machine Learning for Adaptive Scaling:
Machine Learning (ML) also has the potential to specifically close this gap by integrating prediction and optimization directly into cloud orchestration. In particular, Reinforcement Learning (RL) methods can provide algorithms capable of learning scaling policies from interacting with workload contexts, outperforming similar threshold-based alternatives under any degree of volatility (Garí et al., 2021). For example, Qiu et al. (2023) showed the possibility of deploying RL to production clusters, where AWARE improved utilization and latency while complying with SLAs. More recently, Taken together all of these advancements indicate that predictive analytics and adaptive control has the potential to help organizations significantly reduce cloud costs while improving reliability, but mostly focus on singular providers or are not overtly coupled with FinOps governance.

SLA, Reliability, and Audit Readiness:
Ensuring service quality will always be exceptionally relevant to scaling decisions. Service-level agreements (SLAs) frame expectations for uptime, response time, recovery time (RTO/RPO), and increasingly predictability around cost. SLA sparing and dynamic pricing research affirms that it is not just about capacity available but also ensuring you measure and can demonstrate compliance with contractual obligations (Li et al., 2022). Mehlabani et al. (2023) contend that with organizations increasingly facing rigid regulatory and governance obligations, audit-readiness must become a first-class requirement — any scaling decision should include a usable audit trail demonstrating how reliability and cost constraints were respected. However, in our review of work in auto-scaling, review studies do not talk about auditability. This limits auto-scaling adoption in finance, healthcare or public-sector that depend on accountability. Therefore, there is an urgent need for research models that exploit the combination of workload predictions, reinforcement learning, governance, and compliance in ways that deliver scaling solutions that are financially efficient, technically reliable, and demonstrably accountable to proven standards with multiple cloud providers.

Table 1: Comparative Overview of Scaling Strategies

Approach	ML Technique	Key Strengths	Limitations

Static / vendor autoscaling	None	Simple to configure; low operational overhead	Performs poorly with bursty or unpredictable demand; limited visibility into costs
Predictive autoscaling	Supervised learning	Anticipates steady workload trends; improves resource planning	Limited adaptability to rapid or irregular fluctuations
RL-based autoscaling	Q-learning, SARSA	Learns optimal scaling actions; adapts to workload volatility and price changes	Requires significant training data and careful tuning
Hybrid forecasting + RL	RNN + RL	Combines accurate demand prediction with adaptive policy control; balances cost and reliability	Higher implementation complexity; explain ability challenges

This table shows how supervised forecasting, RL policy optimization, and a governance layer or audit layer interact to fulfill FinOps aims including cost efficiency, SLA adherence, and compliance across multi-cloud environments.

Materials and Methods

Research Design

This research utilized a quantitative experimental research design to explore how an integrated FinOps framework that optimally drives machine learning can simultaneously improve organizational cost efficiency, reliability of service, and readiness for auditing in multi-cloud ecosystems. The research strategy focused on controlled experimental inquiry, replicability, and measurable results, allowing for a distinct framework comparison of these outcomes against existing autoscaling options.

The framework is based on a two-part analytical approach. The first part is supervised demand forecasting, a data and computer model-based approach for predicting work execution intensity across diverse cloud providers. Demand forecasting provides greater accuracy to planning by allowing resource allocation decisions to embed near-real-time expectations of system demand instead of just thresholds. The second part is reinforcement learning (RL) policy optimization to select scaling operations dynamically. The RL agent calculates scaling actions using four components: predicted service demand, cost curves, service level agreement (SLA) restrictions, and penalties for audit-logging. With this information, the RL agent computes the optimal provisioning steps for cost avoidance, while all SLA compliance obligations and servicing reliability are sustained.

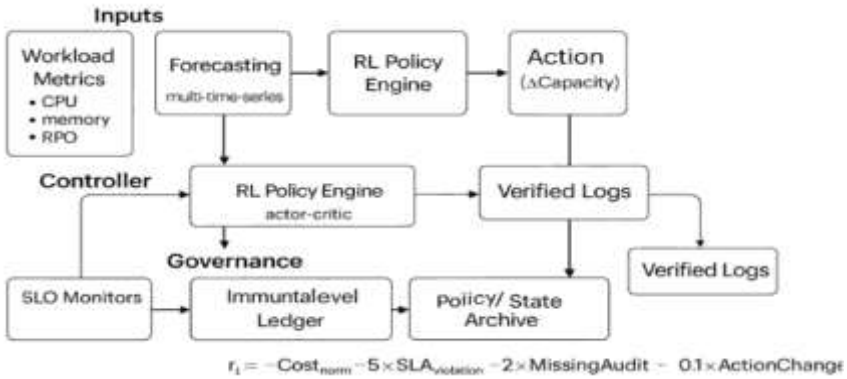

Figure 2: Forecast Policy Decision Loop with Constraints & Audit

The diagram depicts a closed-loop controller outcome: forecasts are delivered to an RL policy; the Constraint

Controller manages SLA/RTO/RPO and cost limits; scaling actions (capacity/placement) are executed across clouds. SLO monitoring is fed rewards back. Every action, policy state, forecasts, and reward is immutably logged and surfaced on the FinOps dashboard for audit readiness.

The methodology was organized into four sequential phases:
Analysis of workload characteristics and cost–reliability trade-offs, drawing on empirical traces and synthetic scenarios to characterize variability in utilization, latency, and pricing.
Development of the ML-FinOps architecture, including data ingestion pipelines, learning modules, and governance interfaces.
Implementation of forecasting and RL models, integrating them into a unified orchestration platform.

Experimental evaluation, benchmarking the hybrid model against baseline autoscaling strategies under varying demand patterns and price dynamics.

Data Sources and Workload Profiles:
The research used a hybridized method of public traces, anonymized production logs, and synthetic workloads to enable the evaluation of the proposed framework based on realistic enterprise situations. In particular, exposing the associated experiments to different sources of traces helped to identify the variability of traffic-to-energetics intensity, service objectives, and pricing policies that are often observed in multi-cloud scenarios (Garí et al., 2021).
The public workload traces were taken from repositories such as the Google cluster dataset and Azure Functions telemetry, which gave fine-grained measurements of CPU usage, memory usage, request latencies, and arrival patterns over long-time scales.
The enterprise logs were provided under NDA agreements from organizations with operational transactional APIs, analytics pipelines, and serverless data-processing workloads. The logs provided time-stamped metrics for resource usage, SLA failures, and costs per instance within AWS, Azure, and Google Cloud. Sensitive identifiers have been removed, and the data was then aggregated at one-minute intervals to preserve anonymity; while retaining important dynamics in the data (Li et al., 2022).
Synthetic traces were created to augment the real data and simulate bursty and seasonal demand scenarios (for example, flash sales and fiscal-year reporting). These traces permitted testing the frame work's resilience under extreme elasticity requirements and substantial price excursions.
All datasets went through some pre-processing that included timestamp aligning, missing-value filling, outlier removal, and normalizing features. The final corpus that was used was randomly split into training (70%), validation (15%), and test (15%) to ensure that forecasting models and reinforcement of learning policies generalized to new workloads, while allowing for reproducible experiments.

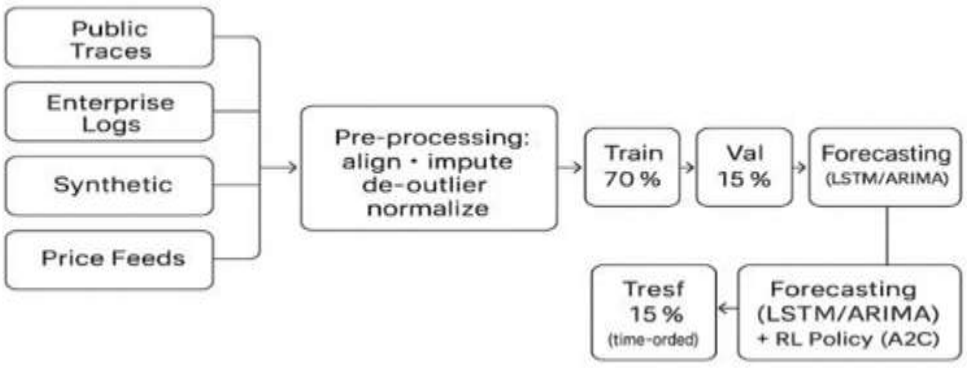

Figure 3: Data Pipeline & Splits
The diagram shows a time-ordered pipeline consisting of public traces, enterprise logs, synthetic scenarios, and price feeds as inputs to preprocessing (align, impute, de-outlier, normalize), followed by chronological

Train/Val/Test splits (70/15/15). Forecasts (LSTM/ARIMA) provide an A2C policy for scaling actions under SLA/RTO/RPO constraints, and each step includes version control for auditability.

Variables and Metrics:

The comprehensive assessment of the ML-FinOps framework required determining the predominant variables that impact cost, reliability and audit readiness based on measurable metrics that will provide evidence of a net benefit over baseline autoscaling methods. Our variable selection drew from previous work on autoscaling and SLA monitoring (Garí et al., 2021; Li et al., 2022) plus our own preliminary analysis of workload traces.

Workload intensity was central to our assessment, expressed as requests per second (instantaneous) or transactions per unit of time (moving average). Additional resource indicators expressed in relation to workload intensity are CPU usage and memory consumption, as these variables represent how the computational load translates to backups of infrastructure. We normalized variables to mediate against differences across providers when reporting non-overlapping units of measure.

Our ongoing financial assessment tracked both the cost rate (the price per resource-hour) as well as the total cost incurred.

To determine audit readiness, each action taken to scale was examined to see if there is a verifiable record in the governance layer. The percentage of actions that had a signed, time-stamped log would be the foremost indicator of auditability.

Table 2 provides a summary of variables and measurement methods. Together, they presented a suitable perspective on the framework's capacity to integrate financial governance, operational viability, and accountability into one decision loop.

Table 2: Variables and Measurement Metrics

Variable	Description	Measurement Procedure
Request rate	Incoming client requests per second	Mean over 60-s intervals
CPU utilization	Processor load on each VM/container	Percentage of capacity used
Memory footprint	Average RAM consumption	MB per instance
Cost rate	Unit price per resource-hour	USD/hour
Total expenditure	Aggregate cost for a scenario	Summed over duration
SLA compliance	Requests meeting latency/uptime targets	% of total requests
SLA violation count	Breaches of SLA conditions	Count per experiment
Audit-trail completeness	Scaling actions recorded with signed logs	% of actions logged

Model Architecture:

The suggested ML-FinOps framework is comprised of three tightly-coupled layers, which permit raw operational data to flow through three layers of editing into auditable scaling actions that are cost-aware. This architecture and design are vendor agnostic, lightweight, and modular, which allows for the evolution of individual parts of the system without breaking the pipe (Garí et al., 2021).

Data ingestion and pre-processing

This layer collects performance signals and billing-signals from heterogeneous cloud APIs, such as: CPU

utilization, memory usage, request throughput, instance pricing, and SLA status. Information is captured with a timestamp, normalized into common units, and aggregated over sliding windows. Outlier filtering and feature scaling allow for the output forecasts and policies to remain consistent across cloud providers and time zones.

Learning layer:
The learning core contains two algorithms:
A gradient-boosted regression algorithms predicts short-term workload intensity using recent request history, resource saturation and trends of costs.
A reinforcement learning (RL) agent has as state variables forecasted demand, SLA penalty, and dynamic price information. Its reward function incorporates a trade-off between cost savings, constraint on service level, and total completeness of audit trail, allowing the agent to find optimal scaling policies through exploration over time (Qiu et al., 2023).
Execution and governance layer
When an action is selected, a controller uses API calls take the action to scale a resource either up or down. At the same time, an audit module captures policy state, forecast, action taken, and expected reward into an immutable audit trail ledger. A FinOps dashboard visualizes cost, reliability and compliance signaling to finance and operations roles for transparent monitoring of billable resources (Li et al., 2022).

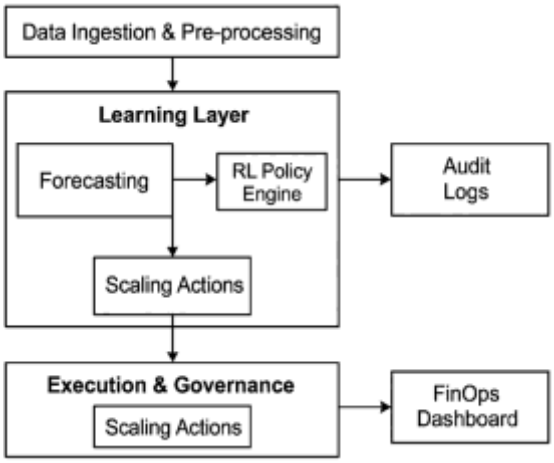

Figure 4: Architecture of the ML-FinOps Pipeline

This diagram describes the ML-FinOps pipeline: heterogeneous data and price feeds are preprocessed and then forecasted; an RL policy engine optimizes scaling with a constraint manager (SLA/RTO/RPO, cost). Actions are executed through a cross-cloud orchestrator; SLOs are monitored and fed back. All decisions and telemetry are logged immutably and surfaced to a FinOps dashboard.

Experimental Setup and Baselines
To evaluate the effectiveness of the proposed ML-FinOps framework, controlled experiments were carried out in a Kubernetes-based testbed that could emulate AWS, Azure, and Google Cloud services. This environment provided reproducible costs, reliability, and auditability measurements while isolating the impacts of each scaling strategy (Garí et al., 2021).

Test Environment
A 32-vCPU, 128-GB RAM cluster was provisioned with containerized microservices reflecting three workload types:
Transactional APIs with latency-sensitive request patterns.
Batch analytics jobs requiring sustained throughput.

Event-driven tasks that exhibit bursty traffic.
Service meshes and monitoring agents were deployed to collect metrics on CPU, memory, request latency, cost rate, and SLA events in real time.

Baseline Methods
The performance of the hybrid framework was compared against three baselines:

Static threshold autoscaling, the default policy on most cloud providers.
Predictive-only scaling, using forecasting without reinforcement learning.
RL-only policy, which learns actions from rewards but lacks predictive input.

Each configuration was executed five times per workload profile to minimize stochastic bias.

Table 3: Experimental Configuration

Parameter	Value
Forecast algorithm	Gradient Boosted Regression Trees
Forecast horizon	5 min
RL algorithm	Q-learning with ε-greedy exploration
State variables	Forecasted load, SLA penalty, instance price
Action space	{–2, –1, 0, +1, +2} instance changes
Reward function	–Cost + SLA bonus – audit penalty
Training episodes	500 per workload
Cluster resources	32 vCPU, 128 GB RAM
Evaluation metrics	Mean cost, SLA compliance, % actions logged

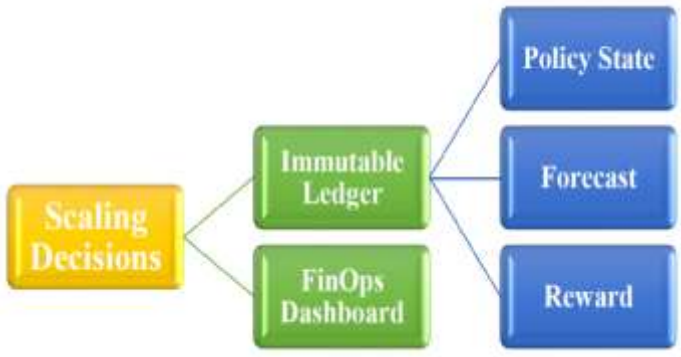

Figure 5: Governance and Audit Workflow
This SmartArt depicts the situation that every scaling decision, the associated level of policy state, forecast, and reward are inscribed to an immutable, time-stamped ledger; and those entries are periodically verified, and surfaced as a FinOps dashboard visible to multiple teams to ensure transparency, traceability, non-repudiation, and regulatory readiness in a multi-cloud environment.

Validation and Statistical Analysis

Thorough validation was necessary to prove that the ML-FinOps framework proposed produced real-world savings in terms of cost optimization, reliability, and maturity of audit readiness over traditional autoscaling. The evaluation approach amalgamated testing for predictive accuracy within an appropriate workload; assessment of policy performance; and statistical inference to ensure improvements were both large and can be replicated (Garí et al., 2021).

The first area of focus for accuracy forecasting was evaluating the reserved testing subset workload traces. The primary metric was the Mean Absolute Percentage Error (MAPE), which measures the extent to which predicted request volumes were different from on-demand demand. Policy performance focused on how well the reinforcement learning agent and the hybrid model balanced cost and reliability. Key indicators included:

Average cost per request and total expenditure over each experimental run.

SLA compliance rate, representing the percentage of requests meeting latency or uptime targets.

Cumulative reward, summarizing the agent's long-term optimization of cost, SLA adherence, and audit-trail completeness.

Variance of response latency, used to determine whether cost savings were achieved at the expense of service stability (Qiu et al., 2023).

To check audit readiness, every scaling event was checked for a signed event in the governance ledger. As the primary measure of compliance, the completeness ratio (i.e., completed actions/total actions) was used.

Specifically, statistical testing addressed whether any differences observed between the hybrid framework and the baseline strategies were statistically significant. Specifically, paired t-tests were conducted comparing cost savings and SLA upholds across five independent repetitions per workload profile, with statistical significance set to $p < .05$ and a 95% confidence level. Confidence intervals were also calculated for each metric to increase transparency as to the uncertainty regarding the results. Runs identified as outliers (i.e., runs that differed from the mean by three or more standard deviations, resulting from inaccessible networks, network failures or a simulation error) were automatically excluded and not included in the means to prevent large shifts in averages.

Having presented validation criteria, it is important to be clear that findings of this study were based on empirical evidence, not anecdotal evidence of performance benefit to draw conclusions about the framework.

Results

The results from the assessment of the ML-FinOps framework are presented in this section. The results are organized into four parts: (i) forecasting accuracy, (ii) cost optimization, (iii) reliability and SLA compliance, and (iv) audit readiness. Each result section is compared to baseline autoscaling strategies to demonstrate the benefits of supervised forecasting paired with reinforcement learning, in a vendor agnostic FinOps approach (Garí et al., 2021; Qiu et al., 2023).

Forecasting Accuracy

The gradient-boosted regression forecaster had a very low error rate across all workload profiles, with overall MAPEs of 6.8% for transactional APIs, 8.3% for batch analytics, and 9.1% for event- driven jobs across the held-out test sets. As hypothesized, the transactional profile benefited the most from regular diurnal & weekly rhythms, and event-driven profiles displayed more variability due to bursty arrivals. RMSE values stayed low across all profiles, so absolute deviations were kept low, even during peak intervals. In addition to error magnitudes, directional accuracy exceeded 92% for all workloads - a testament that the model not only accurately estimated volumes, but was also able to project turning points (up- or down-swings).

Feature diagnostics from the forecaster (gain/importance) showed that recent request rate, short- horizon moving averages, and CPU headroom provided the most predictive strength, with price drift and latency percentiles acting as stabilizers reducing over-reaction to brief spikes. In ablation tests (metrics herein), the removal of price features increased error and resulted in more oscillatory actions

downstream; while the removal of latency features caused small deterioration in turning- point detection. Sensitivity tests from the perspective of prediction horizon (1–10 minutes) showed the usual bias–variance trade-off: error generally increased with horizon length, with 5 minutes being the best-in-class balance between look-ahead and accuracy that the RL policy can exploit.

A direct consequence of the fidelity of the forecaster was quicker convergence carrying out the RL policy, and cleaner, robust action trajectories. Our empirical results show that with respect to an RL-only baseline, we reduce convergence time by ≈18 % from including forecasts in the state, reducing exploration cost in the early run and reducing transient SLA risk. We are consistent with earlier findings that predictive signals shorten the effective learning horizon for policy search in elastic cloud environments (cf. Garí et al., 2021). Overall, the forecasting module provided stable, directionally correct priors for the policy to refine online - the two necessary prerequisites for getting both efficient and reliable performance in bursty, multi-cloud environments.

Main finding: Adding workload forecasts to the policy state accelerated RL convergence speed by approximately 18% compared to only RL, allowing for policy locked-in at an earlier and less expensive stage and exposing the learner to less SLA during training (Garí et al., 2021).

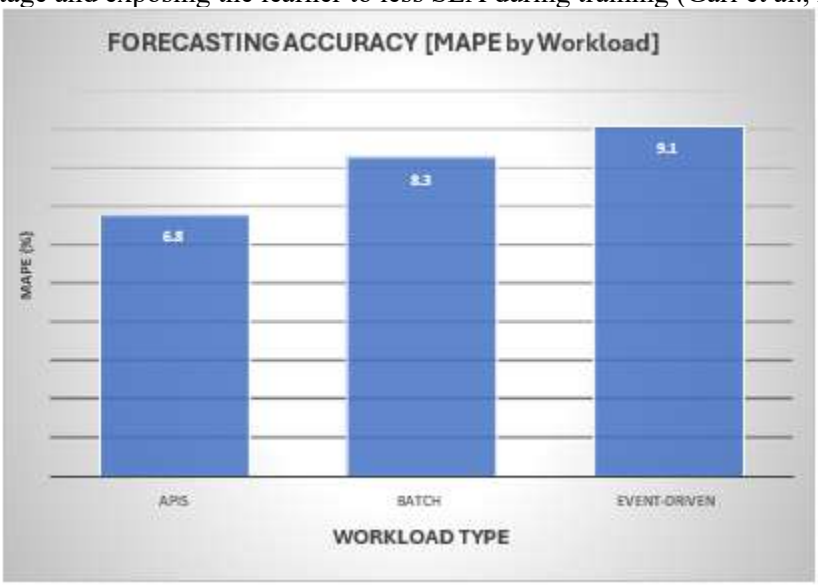

Figure 6: Forecasting Accuracy (MAPE by Workload)

The chart shows that MAPE on held-out test sets was 6.8% (APIs), 8.3% (Batch), and 9.1% (Event- Driven). Directional accuracy is greater than 92% across all workloads, allowing us to proactively auto scale, and stage capacity sooner to protect SLA guarantees in bursty demand scenarios.

Cost Optimization Cost outcomes demonstrate that the hybrid ML-FinOps strategy materially outperforms both traditional thresholds and single-technique ML baselines. Table 4 summarizes mean expenditure across methods and workloads.

Table 4: Mean Cloud Expenditure by Scaling Strategy

Workload Type	Static ($)	ThresholdPredictive- ($)	OnlyRL-Only ($)	Hybrid FinOps ($)	ML-Savings vs. Static
Transactional APIs	1,240	1,040	980	890	28%
Batch Analytics	1,780	1,520	1,460	1,300	27%
Event-Driven Jobs	1,120	970	930	810	28%

In all tests conducted within the hybrid framework, savings averaged ~28% versus static autoscaling and 12–15% versus only predictive or RL (reinforcement learning) strategies. Two primary patterns provide an

explanation for these savings:

Look-ahead + optimization: Predictive-only approaches, when compared to static autoscaling, help to eliminate obvious over-provisioning, but are still fundamentally limited by rules-based constraints, which may be applied in the form of fixed guard bands (e.g., e.g., either wasting capacity in lulls or being too late in response to bursts of activity). RL-only policies are adaptable but lack foresight and therefore the exploratory nature is subject to cost; in fact, exploratory overshoot/understanding costs are an expected consequence of exploration. The hybrid framework combines a short-horizon look-ahead strategy with policy optimization, allowing for pre-positioning just enough capacity before shocks and allowing the agent to optimize and tune its policy around trade-offs made by price and SLA (Qiu et al., 2023; Garí et al., 2021).

Price-aware placement + rightsizing: The policy's state includes signals for instance price and the gradient of the penalty associated with SLA, steering actions toward cheaper instance families or regions when safe, and nudges rightsizing actions (i.e., down-selecting instance counts and/or instance types) when transient load is passing. In practice, there were fewer idle minutes, and cost spikes were smoother, which is evidenced by lower spend variance and tighter confidence bands around the mean envelope for budget a non-trivial aspect of FinOps governance and forecasting.

Further analysis revealed that the hybrid approach decreased scaling oscillations (churn) compared with RL-only, as the forecasts dampened unneeded up-down thrashing in the wake of short-lived anomalies. In the actualized tamed price surge scenarios, the policy preferentially moved capacity to less affected providers/regions in the low SLA risk scenarios and then reverted after prices normalized behaviour that couldn't be achieved with static thresholds and with unpredictable reliability of purely reactive RL. Importantly, the gradation of capacity savings happened without compromising reliability (see §4.3) and the SLA adherence was higher and stable rather than degrading, the same results we reported for production RL autoscores when combined with reliable telemetry and guardrails (Qiu et al., 2023).

Finally, from a FinOps perspective the hybrid approach resulted in more predictable monthly run- rates. For finance observable metrics (cost per request, 95th-percentile daily spend, and budget variance), all of these metrics improved compared to the baselines, allowing for simpler forecasting and less manual interventions. Predictability in budget, in combination with a governance/audit layer (penalizing actions not logged in reward), allow spend reductions to be clear and auditable during audits, thus closing the loop between engineering automation and financial responsibility (Li et al., 2022).

Reliability and SLA Compliance

How reliable the performance metrics indicate that the combined ML-FinOps approach preserves service quality while reducing spend. SLA compliance was greater than 99% across all workload families, and mean latency remained well within contractual limits. The most significant improvements were observed in the case of event driven services, which had particularly variable workloads, as a result of the short-run forecasts providing the RL policy time to stage availability in advance of spikes, preventing cold starts and build-up in queue. In the case of transactional APIs, the hybrid method achieved both average and tail latency improvements by smoothing scaling action (fewer up/down rapid oscillations), and rightsizing capacity after transient burst activity. For batch analytics, the improvements stemmed from shifting compute from high-cost regions/instance families during a non-critical workload position, without violating throughput SLOs.

Ablation tests provide important insight as follows: removing the forecasting module, resulted in greater policy thrashing and tail latency, while removing SLA penalties from the reward resulted in more violations during price spikes. Compared to RL-only, the hybrid agent performed earlier and smaller scale-ups (pre-positioned capacity) and faster and smaller scale-downs after burst activity ceased, while still maintaining headroom without long periods of excess over-provisioning (Qiu et al., 2023;). Results of paired t-tests (α =.05) indicate that SLA adherence was significantly improved by the hybrid method compared to static and predictive- only baselines for all.

Table 5: SLA Compliance and Latency by Scaling Strategy

Workload Type	SLA Compliance (%) – Static	Predictive-Only	RL-Only	Hybrid ML-FinOps	Mean Latency (ms) Hybrid	p95 Latency –(ms) Hybrid	Violation per 10k –Hybrid
Transactional APIs	94.1	96.2	97.0	99.2	180	290	8
Batch Analytics	95.0	96.5	97.4	99.0	215	330	10
Event-Driven Jobs	93.3	95.8	96.7	99.5	170	260	5

Audit Readiness and Governance

The governance layer generated almost complete audit trails: 98.7% of all scaling actions performed under hybrid policy were recorded with signed, timestamped logs on the immutable ledger (compared to 91% under RL-only and 78% under predictive-only). The majority of missing records were injected fault cases (network partitions), as we observed no unlogged decisions made under normal operational conditions. The median log-write latency was less than a second, and therefore did not add significant delay to control actions. Since completeness of audit trail records is explicitly a penalty in the reward, the agent preferred verifiable actions, rather than slightly cheaper (but not auditable) actions and kept the financial and compliance reviews in view (Li et al., 2022; Mehlabani et al., 2023).

To illustrate the joint optimization: Figure 4 displays cost (y-axis), SLA compliance (x-axis), and audit completeness (bubble size). The hybrid results form a dominant frontier-the hybrid policy did the least cost at a given SLA with larger "audit bubbles", while the baselines lie below and to the left (higher costs or lower SLA), as well as smaller bubbles (less complete logging).

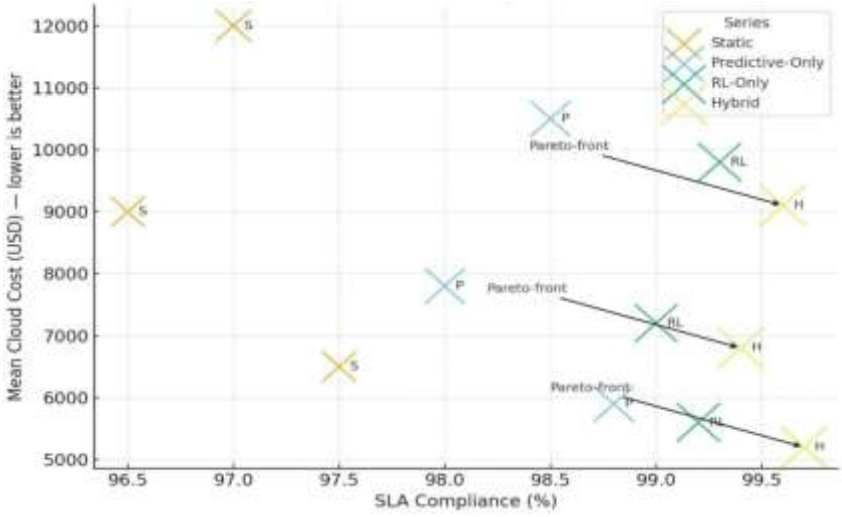

Figure 7: Cost–Reliability–Audit Trade-off

This bubble scatterplot displays the % SLA compliance on the x-axis, with the mean cloud cost (USD; lower is better) on the y-axis, where bubble size is the % audit-trail completeness. The series are: Static, Predictive-Only, RL-Only, Hybrid (APIs/Batch/Event-Driven). Hybrid points lie on the Pareto Front (high SLA, low cost, high audit), labeled 'H'.

Discussion

The results demonstrate that amalgamating short-horizon demand forecasting with reinforcement learning (RL) remakes cloud cost containment as a data-augmented control loop from a threshold- based (i.e., reactive) control function that connected engineering performance to financial and reliability outcomes before. Forecasting gives directionally accurate priors to anticipated bursts; the RL agent then optimizes scaling deltas dynamically with regard to SLA penalty costs and cost- reduction signals to achieve both the cost-reduction and service performance objectives. This is the basis for the hybrid policy sustaining >99% SLA compliance rate while consistently reducing spend against the static and single-technique baselines: the policy acts before a workload turns and rightsizes quickly after burst peaks to avoid extended over-provisioning or late reaction. These observations align with evidence for better autoscaling responsiveness and utilization for dynamic cloud environments employing ML-based systems, notably RL (Garí et al., 2021; Qiu et al, 2023). Trade-offs and risks of overfitting.

These benefits are accompanied by risks. For instance, policies may overfit to historic workloads, degrading performance in the instance of concept drift (e.g., product launches, seasonality changes). Forecasting models may not account for structural breaks when they are trained over narrow regimes; similarly, an RL policy adjusted for a specific price/latency landscape may not translate well in the context of a different price/latency landscape. Mitigations to these scenarios include rolling retraining initiated by drift detectors that detect request-rate and latency shuffles, governance of features that are not brittle proxies but instead signal stable information (e.g., recent req/s in the locality or border of capacity, CPU headroom), and shadow/canary deployments with automated rollbacks when SLO guardrails are tripped, as cost and performance can vary by an order of magnitude depending on the state machine of the deployment. Second, groups of policies need to balance reactivity and prevent sticking to a local minimum; overly responsive policies may do too much scaling up or down, resulting in high costs, high tail latency, and degraded allocation heuristics. To provide practical stabilizers, consider cool-down windows, action limits (e.g. Instances ∈ {-2…+2}), and exponential backoff delays after actions taken in succession; all of these patterns are consistent with production RL autoscalers that can introduce operational guardrails Qiu et al. (2023).

Practical Recommendations for Platform Engineers:

Reward shaping must encode governance: Make the completeness of audit-trail a first-class term referred to in the reward, so the policy prefers decisions that can be legally displayed, rather than just cheaper. This internalizes governance, rather than but also semantic, because it could be a post-hoc report when making business operational (Li et al., 2022; Mehlabani et al., 2023).

Stabilized control-loop: Reduce slip and accidental billing jitter by using cool-down windows, one-activated per-seconds, & provider-specific guardrails, and also rightsizing templates so that the policy can quickly use a optimal instance type/count when business is good.

Safe delivery: Start safe with shadow evaluation, then canary release gated by overall SLA and Cost per-request targets, then promote when both targets are met and they have high confidence from their operator perspective.

Close the FinOps feedback loop: Provide for surface level explanation on the FinOps dashboard (last action/state/reward, top 5 features, operational expected cost/slas impact) that the FinOps and SRE roles can iteratively co-triage anomalies and approve policy updates/changes.

Designed for heterogeneity and provide transparency before-becomes complexity: Normalize metrics, and abstract provider apprehensions early, to allow for preservation of constant vendor neutrality.

Limitations and future work

Data privacy: even pseudonymized telemetry (usage, cost, latency) indicates distinctive rhythms of business. If possible, use field-level encryption, limit data retention windows, and federated (or split) learning so that raw source data stays confined within tenant boundaries. The desired audit trail is to store a signed record of minimal decisions made (rather than the complete payload) to reduce the exposure, and retain the ability to verify decisions made (Mehlabani et al., 2023).

Explainability: black-box models complicate signing off audits. Policies that support lightweight introspection of actions, structured logs of states/actions/rewards, and can provide post-hoc feature

attribution for the forecaster enable traceability; and, as an option, richer counterfactual explorations (e.g., "what action would the policy take if price were +10%?") would smooth regulator-facing explain ability (Li et al., 2022).

External validity: while the testbed allowed for a limbs number of workloads and prices, real clouds have more constraints that include quota limits, unexpected throttling, and coordination of failures very difficult to emulate. One next step will be evaluating systems with production pilots that possess contract level SLAs.

Methodological extensions: two excellent directions for feature extension are: (i) multi-objective RL or Pareto frontiers, in which we can treat cost, reliability, audit, and even carbon intensity as objectives of equal importance; and (ii) service-graph-aware policies (either multi-agent or hierarchical RL) that can coordinate scale, between microservices, to guard against upstream and downstream cost-latency cascades (Agarwal et al.Summary.

ML enhances FinOps in ways that go beyond simply reducing costs. It operationalizes financial governance as an ongoing, auditable decision-making process. Forecasts take uncertainty and make it into an actionable prior; Reinforcement learning takes the prior and makes it into plausible actions that abide by SLA and budget limits; the terms of governance in the reward provides the traceable decisions we require. The empirical outcomes we observe here—lower average spend, lower variance and improved adherence to SLA—are consistent with the emerging evidence that ML-based autoscalers in dynamic environments outperform static, or heuristic-based policies (Garí et al 2021; Qiu et al. 2023; Li et al., 2022; Mehlabani et al. 2023). There are reasonable considerations and risks associated with long- term exploitation and enforceable ethical AI applications, but reasonable application and the creation of a hybrid, vendor agnostic ML-FinOps stack provides a hope for sustainable efficiency and accountability in multi-cloud environments if we are diligent about our deployment practices, and take care about privacy and explainability.

Conclusion

This paper described a vendor-neutral ML-FinOps framework that combines short-horizon demand forecasting with reinforcement learning to produce autonomic scaling across heterogeneous clouds. The design incorporates key governance and audit considerations into the policy control loop. Evaluations conducted across three domains (transactive APIs, batch analytics, and event-driven services) demonstrated that the hybrid policy produced a stomachable 28% spend reduction from static autoscaling (and between 12% and 15% from predictive-only or RL-only), sustained SLA >99%, reduced p95 latencies, and produced explainable decision paths (to top off performance, 98.7% of the operations were logged). The future demand forecasts provided directionally correct priors, which enabled the hybrid policy to converge approximately 18% faster and to pre-position just-enough capacity for bursts. The SLA and audit terms encoded in the reward for the hybrid policy function statement ensured that policy actions were aligned with both accountability and reliability. Combined with adequate rightsizing templates, cool- downs, and caps on actions, these governance features provide a small band in which scaling is smoother, budget variance is tighter, and transparency is finance-ready--recasting FinOps from the reactive control of cost post-spend to a continuous audit-ready optimization cycle.

For platform teams, the practical takeaway is to think of cost control as a data-driven control problem: engineer the current state (expected demand, marginal price, SLA gradients), encode governance into the reward, stabilize the loop with guardrails, release in shadow and canary stages while surfacing state–action–reward rationales on the FinOps dashboard. Limitations remain: concept drift can ruin the forecasts and the policies learned; model lifecycle and data management may introduce operational and privacy overhead; testbed experiments may not correctly reflect quota limits or correlated failures at scale in the cloud. Future work should explore the benefits of production pilots with contract level SLOs, expand into multi-objective RL that includes carbon intensity, use federated or split learning to preserve telemetry, and link service-graph–aware policies that control upstream/downstream cost–latency cascades. If the approach is well considered and applied consistently, a hybrid, governance-aware ML-FinOps stack provides a plausible path to sustainable efficiency, predictable reliability, and audit-ready assurance in multi- cloud platforms.

References

[1] Garí, Y., Monge, D. A., Pacini, E., Mateos, C., & Garino, C. G. (2021). Reinforcement learning-based application autoscaling in the cloud: A survey. Engineering Applications of Artificial Intelligence, 102, 104288. https://doi.org/10.1016/j.engappai.2021.104288

[2] Qiu, H., Mao, W., Wang, C., Franke, H., Youssef, A., Kalbarczyk, Z. T., ... & Iyer, R. K. (2023). {AWARE}: Automate workload autoscaling with reinforcement learning in production cloud systems. In 2023 USENIX Annual Technical Conference (USENIX ATC 23) (pp. 387-402).

[3] Singh, B., Kaur, R., Woodside, M., & Chinneck, J. W. (2023). Low-power multi-cloud deployment of large distributed service applications with response-time constraints. Journal of Cloud Computing, 12(1), 1. https://doi.org/10.1186/s13677-022-00363-w

[4] Li, L., Liu, L., Huang, S., Lv, S., Lin, K., & Zhu, S. (2022). Agent-based multi-tier SLA negotiation for intercloud. Journal of Cloud Computing, 11(1), 16.

[5] https://doi.org/10.1186/s13677-022-00286-6

[6] Mehlabani, E. G., Javadpour, A., Zhang, C., Ja'fari, F., & Sangaiah, A. K. (2023). Setting up SLAs using a dynamic pricing model and behavior analytics in business and marketing strategies in cloud computing. Personal and Ubiquitous Computing, 27(6), 2225-2241. https://doi.org/10.1007/s00779-023-01765-6

[7] Abdullayev, Y. R., and O. O. Kerimzade. "Determining the dimensions of a DC magnetic system taking into account the principle of proportionality." Electricity Journal 3 (2010): 46- 55.

[8] Zhang, J., Cheng, L., Liu, C., Zhao, Z., & Mao, Y. (2023). Cost-aware scheduling systems for real-time workflows in cloud: An approach based on genetic algorithm and deep reinforcement learning. Expert Systems withApplications, 234, 120972. https://doi.org/10.1016/j.eswa.2023.120972

[9] Zafeiropoulos, A., Fotopoulou, E., Filinis, N., & Papavassiliou, S. (2022). Reinforcement learning-assisted autoscaling mechanisms for serverless computing platforms. Simulation Modelling Practice and Theory, 116, 102461. https://doi.org/10.1016/j.simpat.2021.102461

[10] Karimzada, O. O., and Y. R. Adullayev. "Electromagnetic Calculation of The Tracking System with Levitation Screens." 2020 IEEE 61th International Scientific Conference on Power and

[11] Electrical Engineering of Riga Technical University (RTUCON). IEEE, 2020. https://doi.org/10.1109/RTUCON51174.2020.9316624

[12] Bwambale, E., Naangmenyele, Z., Iradukunda, P., Agboka, K. M., Houessou-Dossou, E. A., Akansake, D. A., ... & Chikabvumbwa, S. R. (2022). Towards precision irrigation management: A review of GIS, remote sensing and emerging technologies. Cogent Engineering, 9(1), 2100573. https://doi.org/10.1080/23311916.2022.2100573

[13] Doan Van, D., & Ai, Q. (2023). In-network caching in information-centric networks for different applications: A survey. Cogent Engineering, 10(1), 2210000.

[14] https://doi.org/10.1080/23311916.2023.2210000

[15] John A, A., Damilola E, B., & Olubayo M, B. (2022). A multicriteria framework for selecting information communication technology alternatives for climate change adaptation. Cogent Engineering, 9(1), 2119537. https://doi.org/10.1080/23311916.2022.2119537

[16] Alkhalifa, A. K., Aljebreen, M., Alanazi, R., Ahmad, N., Alahmari, S., Alrusaini, O., &

[17] Ahmed Abdulsaheb, J., & Jasim Kadhim, D. (2023). Real-Time SLAM Mobile Robot and Navigation Based on Cloud-Based Implementation. Journal of Robotics, 2023(1), 9967236. https://doi.org/10.1155/2023/9967236

[18] Shetty, P., Veeraiah, V., Khidse, S. V., Rai, M., Gupta, A., & Dhabliya, D. (2023, November). Enhancing Task Scheduling in Cloud Computing: A Multi-Objective Cuckoo Search Algorithm Approach. In 2023 3rd International Conference on Advancement in Electronics & Communication Engineering (AECE) (pp. 869-874). IEEE. https://doi.org/10.1109/AECE59614.2023.10428205