Intelligent Cost Optimization in Cloud Migration Using Machine Learning

Pratik Jadhav, Rohan Katkar, Karan Bhongale, Vedant Chaudhari, Prashant Chordiya
Student, Department of Master of Computer Applications and Management.
Assoc. Prof., Department of Master of Computer Applications and Management.
Dr. D. Y. Patil Institute of Master of Computer Applications and Management, Akurdi, Pune.

Abstract

Cloud migration gives organizations scalable computing without large upfront hardware investment, but it can also raise operational expenditure when resources are over-provisioned, workloads are poorly estimated, or pricing models are chosen carelessly. Existing practice often relies on static thresholds, manual capacity planning and periodic cost reviews, which struggle with workloads that change quickly or behave non-linearly. Earlier research has studied workload prediction, autoscaling, virtual-machine consolidation and cost-aware provisioning, but these topics are usually examined separately, and an accurate forecast does not by itself prove that cloud spending falls.

To address this issue, this paper proposes a machine-learning framework that connects workload forecasting with a constrained decision layer. Telemetry such as CPU, memory, request rate, latency, storage activity, network traffic, instance count and billing data is used to predict future demand. A transparent Safe Cost Optimization Score (SCOS) then ranks candidate actions such as resizing, scaling, consolidation, scheduling, pricing-plan changes and decommissioning, while hard limits on latency, availability and operational risk keep recommendations safe.

This research does not present new experimental findings. The studies discussed are described qualitatively from their published scope, and the proposed SCOS framework is a new concept that requires further testing and validation. The overall analysis suggests that forecasting accuracy is only one input to a good cost decision; cost, performance, reliability and governance must be evaluated together.

The main contribution is a practical, reproducible framework for evaluating and improving cloud-cost decisions during and after migration. Rather than minimizing expenditure alone, the approach aims to deliver financially beneficial, explainable and operationally safe resource decisions.

Keywords: cloud migration; cloud cost optimization; machine learning; workload forecasting; predictive autoscaling; resource provisioning; FinOps; constrained optimization

1. Introduction

Cloud computing has changed how organizations acquire, deploy and operate infrastructure [1], [2]. Instead of buying fixed-capacity hardware, organizations obtain compute, storage and networking on demand [14]. This supports elasticity and rapid deployment, but it also turns infrastructure spending from mostly capital expenditure into usage-dependent operational expenditure [1].

Cloud migration is more than moving applications from an on-premises data center. It involves choosing deployment models, sizing resources, estimating migration costs, selecting pricing options and preserving performance during the transition [27]. Migration may temporarily require duplicate infrastructure, data-transfer services, backups and test environments, so migration cost should be analysed separately from long-term operating cost.

A major difficulty is the mismatch between provisioned capacity and actual demand [1], [15]. Organizations may reserve excess capacity to avoid slowdowns at peak, and idle capacity keeps generating charges at quiet times. Aggressive cost cutting causes the opposite problem: under-provisioning, higher latency and service-level-objective (SLO) violations. Traditional tools use static rules, for example adding instances when CPU exceeds 70% and removing them below 30% [15]. Such rules are simple to interpret but do not model future demand, so they may react late to sudden spikes or keep unnecessary capacity [6], [8].

However, the benefit of machine learning is not the same in every setting. Forecasting models can predict demand in advance [5], [18], but a model may be accurate and still trigger excessive scaling, ignore data-transfer charges or recommend configurations that break latency targets [24]. This suggests that predicting demand does not automatically mean reducing cost.

To better understand these effects, this paper brings together cloud economics [1], [2], resource management [27], workload prediction [5]–[7], autoscaling [8], [15] and machine learning [20]. The central idea is that cost-optimization systems should not be judged only by prediction accuracy or by cost savings in isolation. They should also be evaluated on performance, reliability, operational risk and governance.

2. Research Problem, Questions, and Objectives

Most cloud-cost practice today depends on utilization thresholds, manual right-sizing and monthly bill reviews [15], [27]. These approaches react after demand has already changed [6]. Telemetry and billing data are often held in separate systems, so decisions are made with incomplete information. Cost minimization without explicit constraints can also recommend over-consolidation that hurts latency or availability [4], [19].

Another major concern is that existing research tends to study forecasting, provisioning, autoscaling and pricing separately [15], [20], [27]. This makes it difficult to judge how a forecast turns into a financial decision. Therefore, a more balanced approach is needed that evaluates cost decisions together with their performance and risk effects.

Research Questions

RQ1: How do reactive threshold policies compare with prediction-driven policies in provisioning cost and SLO compliance during and after migration? [6], [8], [15]

RQ2: Which machine-learning models (interpretable, ensemble, time-series, deep learning) forecast cloud demand best relative to their complexity? [7], [18], [28]

RQ3: Can a constrained optimization layer convert forecasts into lower cost without significantly increasing SLO violations or operational risk? [10], [11]

RQ4: How can workload drift be detected and handled so that models remain reliable over time? [26]

Research Objectives

1. To understand how workload forecasting, autoscaling and cost-aware provisioning contribute to cloud cost control [8], [10], [12].
2. To identify measurable factors for evaluating the cost, performance and risk of resource decisions [24].
3. To propose a transparent scoring and constrained decision method that does not focus on cost alone [11].
4. To define a clear and reproducible validation approach, including chronological data splitting [25].
5. To examine the limitations, ethical concerns and safeguards of automated cloud-resource decisions [9], [21].

3. Literature Synthesis

3.1 Cloud Computing and Migration Economics

Armbrust et al. [1] describe cloud computing as shared, configurable resources delivered as services, with an economic benefit that depends on how demand relates to provisioning. Buyya et al. [2] emphasise utility, pricing, service-level agreements and resource allocation. Broader surveys of the field [14], [27] reach a similar conclusion about the importance of resource management challenges. These works show that the cheapest unit price is not always the cheapest configuration: a smaller virtual machine may need more instances to meet throughput, while a larger one may cost more per hour but raise utilization. Total workload cost, rather than isolated unit prices, should drive decisions. Although these studies explain the economics well, they do not provide a predictive framework for individual migration decisions.

3.2 Cloud Resource Management and Consolidation

Calheiros et al. [3] introduced CloudSim, which lets researchers evaluate provisioning and scheduling policies before deployment. Beloglazov et al. [4] surveyed energy-efficient data centers and showed that consolidation can improve efficiency, but consolidation must account for workload volatility: if several workloads spike together, heavy consolidation can cause contention and degraded service. Adaptive heuristics for dynamic virtual-machine consolidation [19] likewise trade energy savings against performance. These studies motivate evaluating consolidation under explicit performance constraints.

3.3 Workload Prediction

Di et al. [5] developed a Bayesian method for host-load prediction on a large Google cluster and showed that short-term cloud load is hard to predict because demand fluctuates rapidly. Gong et al. [6] introduced PRESS, which predicts demand and scales resources before it rises, reducing the delay of launching capacity. Islam et al. [7] studied empirical prediction models for adaptive provisioning and showed that application-resource relationships can be learned

from historical observations. Related predictive-scaling work includes CloudScale [17], which combines demand prediction with scaling for multi-tenant systems, and the work of Roy et al. [16] on predictive models for autoscaling. Calheiros et al. [18] showed that ARIMA-based workload prediction can be linked to application quality of service. These works support the use of time-stamped telemetry, but they also imply that prediction models must be evaluated with time-aware validation, because randomly mixing earlier and later observations lets future information leak into training [25].

3.3.1 Surveys of Elasticity

Al-Dhuraibi et al. [13] review elasticity in cloud computing and its open challenges, including the need to match different models to different workload and resource types. Surveys of auto-scaling techniques [15] and of machine-learning methods for resource provisioning [20] raise the same point. A model trained on CPU-intensive applications may not generalize to memory- or network-intensive ones.

3.4 Autoscaling and Reinforcement Learning

Gandhi et al. [8] treat autoscaling as a control problem in which scaling decisions must balance response time, resource-launch delay and workload variation, rather than looking at CPU utilization alone. Mao et al. [9] applied deep reinforcement learning to resource management, showing that an agent can learn policies from system states and rewards. Surveys of reinforcement-learning autoscaling [15], [20] report promising adaptive behaviour but also challenges in training stability, exploration risk, reward design and deployment safety. Production autoscalers such as Autopilot [21] show that practical systems must also handle safety and operational constraints. For this reason, the proposed framework evaluates reinforcement learning first in simulation or recommendation mode rather than direct control of production infrastructure.

3.5 Cost-Aware Provisioning

Chaisiri et al. [10] formulated provisioning as a cost optimization problem over purchasing options under uncertain demand, showing that pricing commitments and capacity planning should be evaluated jointly. Cohen et al. [11] studied overcommitment with chance constraints and showed that controlled overcommitment can reduce cost while keeping the probability of capacity violations acceptable. Cortez et al. [12] presented Resource Central, a production system that forecasts workload behaviour to support resource-management decisions, highlighting the importance of practical deployment. Large-scale cluster managers such as Borg [22] illustrate the same operational context. Many of these studies use different datasets, metrics and baselines, making direct comparison difficult.

4. Research Gap and Novelty

Research Gap

Existing research shows substantial progress in forecasting, autoscaling, consolidation and cost-aware provisioning. However, these areas are mostly studied separately, and six gaps remain:

- **Prediction versus action:** many studies predict CPU, memory or request rate without connecting the forecast to an explicit financial decision [5]–[7].
- **Reactive management:** many operational policies respond only after utilization has changed, which is unsuitable when launch delays are significant [8], [15].
- **Limited transferability:** providers differ in instance families, pricing, regions and commitment models, so a model validated on one platform may not generalize [13], [20].
- **Cost without SLO constraints:** minimizing cost alone can recommend excessive consolidation or aggressive scale-in [4], [19].
- **Workload and model drift:** seasonality, releases, traffic growth and price changes degrade static models [26].
- **Inconsistent evaluation:** a lower prediction error does not guarantee lower spending if forecasts cause unnecessary scaling [24].

Therefore, the main research gap is not simply the need for another forecasting model. There is a need for a common evaluation framework that considers cost, performance, reliability, risk and governance together and explains how each factor influences the final decision.

Novel Contribution

To address this gap, this paper proposes a unified pipeline that connects forecasting, cost estimation and constrained optimization, together with the Safe Cost Optimization Score (SCOS), a multi-objective score for ranking candidate resource actions. Unlike approaches that minimize cost alone [10], SCOS rewards savings and performance preservation while penalizing migration or reconfiguration cost and operational risk. At this stage, SCOS is a proposed research framework rather than a tested production algorithm. Further experiments will be required to measure its effectiveness.

5. Proposed Framework and Machine-Learning Method

5.1 Inputs and Feature Groups

The proposed system uses four groups of inputs:

(a) Workload Features: CPU utilization, memory utilization, request rate, latency, error rate, storage I/O, network traffic and active instance or container count, tagged with a workload identifier [12], [23].

(b) Cost Features: billing records, pricing-plan information, data-transfer volume, reserved or spot usage and observed cost per transaction [10].

(c) Temporal Features: lagged utilization, rolling means and standard deviations, peak-to-average ratio, hour of day, day of week, request growth rate, and recent scaling or deployment events [7], [18].

(d) Constraint Signals: SLO targets, capacity limits, availability requirements, security and governance rules, and commitment periods [11].

A critical safeguard is that a decision at a given time must not use future information such as the final daily bill, a later utilization value or a future scaling event [25].

5.2 Machine-Learning Components

Forecasting: linear regression as an interpretable baseline, autoregressive and exponential-smoothing models [18], random forest, gradient boosting, support-vector regression, LSTM and GRU [28], and hybrid statistical-ML models [20].

Anomaly and Idle-Resource Detection: statistical or learned behaviour patterns to flag idle or inefficient resources and cost anomalies [12].

Constrained Optimization: linear or mixed-integer programming [10], heuristic search [19], particle-swarm optimization or reinforcement learning [9], chosen by problem size.

Safety Layer: a rule-based layer that blocks actions exceeding predefined risk limits [11], [21], with human approval for high-impact changes such as long-term commitments or provider migration.

The framework begins with simple, interpretable models. A complex deep-learning model is selected only if it gives measurable improvement over simpler ones in accuracy, cost, stability or reliability.

5.3 SCOS Definition

For a candidate action a at decision time t, the Safe Cost Optimization Score is defined as:

$$SCOS(a) = wC \times Save(a) + wP \times Perf(a) + wU \times Util(a) - wM \times Mig(a) - wR \times Risk(a)$$

The five components are:

1. **Save (Cost Saving):** expected cost reduction relative to the current configuration, normalised to 0–1.
2. **Perf (Performance Preservation):** one minus the predicted probability of an SLO violation.
3. **Util (Utilization Efficiency):** closeness of expected utilization to a target band.
4. **Mig (Migration or Reconfiguration Cost):** one-off cost of the action, including data transfer, downtime and rollback.
5. **Risk (Operational Risk):** capacity, dependency, commitment and compliance risk.

The weights must be non-negative and sum to 1: wC + wP + wU + wM + wR = 1. They should be fixed before the main evaluation and tested under different conditions. SCOS ranks only actions that already satisfy the hard constraints in Section 6.1; it never overrides them.

5.4 Plan-Level Stability

Evaluating single actions is not enough, because a sequence of good individual actions can still cause constant scaling [8]. Let π be a sequence of actions over T decision intervals and let N(π) be the number of scaling or reconfiguration actions. A stability penalty is defined as:

$$Churn(\pi) = N(\pi) / T$$

The overall plan objective is:

$$J(\pi) = \Sigma_t SCOS(a_t) - \mu \times Churn(\pi)$$

where μ controls how strongly frequent changes are penalized. A "no action" option is always available when forecast uncertainty is too high.

5.5 Model Training and Evaluation

Models are trained on time-stamped data, labelled by workload, using a chronological train-validation-test split and rolling-origin validation [25]. Calibration checks and prediction-interval coverage are used when uncertainty estimates exist. Three strategies will be compared:

Approach A – Reactive Threshold Scaling: CPU- or memory-threshold autoscaling and fixed-capacity provisioning [15].

Approach B – Forecast-Only Provisioning: forecasts used directly to size capacity, without constrained optimization [6], [17], [18].

Approach C – SCOS-Based Constrained Optimization: forecasts passed through the SCOS and constraint layer.

Manual right-sizing and non-predictive cost optimization are included as additional baselines. Evaluation measures are listed in Section 9.

6. Mathematical Formulation

6.1 Constrained Decision Model

Let the observed workload state at time t be X_t = [u_CPU, u_MEM, q, l, i_storage, n_network, v, c], representing CPU utilization, memory utilization, request rate, latency, storage I/O, network traffic, active instances and observed cost. A forecasting model f estimates future demand at horizon h [5]–[7]:

$$\hat{Y}(t+h) = f(X_t, X_{t-1}, X_{t-2}, \ldots)$$

The optimizer selects an action a_t from a feasible action set A (resize, scale out or in, change container limits, consolidate, reschedule, change pricing plan, decommission, or do nothing) [10], [11]:

$$\text{minimize } E[C(a_t)] \text{ over } a_t \in A$$

$$\text{subject to } L(a_t) \leq L_max,\ Av(a_t) \geq A_min,\ R(a_t) \leq R_max$$

where C is expected cost, L is latency, Av is availability and R is capacity or operational risk. Intelligent cost optimization is therefore not the same as reducing machine count: the best action may add capacity temporarily if the performance benefit exceeds the added cost.

6.2 Forecasting Baselines

As an interpretable baseline, linear regression models future demand as:

$$\hat{y} = \beta_0 + \beta^T x$$

where x is the feature vector, β are the coefficients and β_0 is the intercept. Autoregressive and exponential-smoothing models serve as time-series baselines [18], and tree-based and recurrent models [28] are compared against them.

6.3 Forecast Error Metrics

With actual values y_t, forecasts $\hat{y}_t$ and n observations [24]:

$$MAE = (1/n)\ \Sigma\ |y_t - \hat{y}_t|$$

$$RMSE = \sqrt{[(1/n)\ \Sigma\ (y_t - \hat{y}_t)^2]}$$

$$MAPE = (100/n)\ \Sigma\ |(y_t - \hat{y}_t)\ /\ y_t|$$

These measure model quality only. They do not show whether spending falls, which is why Section 9 adds cost-level metrics.

6.4 Cost and Over/Under-Provisioning Measures

$$Cost\ per\ request = Total\ cost\ /\ Total\ requests$$

$$Over\text{-}provisioned\ capacity = \Sigma\ max(0,\ Provisioned - Required)$$

$$Under\text{-}provisioned\ capacity = \Sigma\ max(0,\ Required - Provisioned)$$

$$SLO\ violation\ rate = Intervals\ violating\ latency\ or\ availability\ targets\ /\ Total\ intervals$$

6.5 Weight Configuration for SCOS

To maintain a balanced score, each weight is non-negative and all weights sum to 1. As an initial configuration for future experiments:

SCOS Component	Weight
Cost Saving (wC)	0.30
Performance Preservation (wP)	0.25
Utilization Efficiency (wU)	0.15
Migration / Reconfiguration Cost (wM)	0.10
Operational Risk (wR)	0.20
Total	1.00

These are only proposed starting weights. They have not been obtained through empirical estimation or real-world testing, and their suitability must be examined through sensitivity analysis and stakeholder feedback before they are considered reliable.

7. Evidence-Based Findings

This paper presents an evidence-based synthesis and a proposed framework. It does not pool raw datasets and does not claim to have trained or tested a new model. No savings or accuracy figures are reported, because none have been measured. Table 1 summarises what the cited studies contribute, described qualitatively.

Table 1. Summary of cited studies and their relevance to the proposed framework

Study area	Representative work	Main contribution	Relevance to this framework
Migration economics	Armbrust et al. [1]; Buyya et al. [2]	Pay-per-use economics, utility and SLA-based allocation	Total cost, not unit price, drives decisions

Study area	Representative work	Main contribution	Relevance to this framework
Simulation	Calheiros et al. [3]	CloudSim for testing provisioning policies	Safe offline evaluation of policies
Consolidation	Beloglazov et al. [4]	Efficiency gains from consolidation under QoS limits	Consolidation needs explicit constraints
Workload prediction	Di et al. [5]; Islam et al. [7]; Gong et al. [6]	Short-term prediction and predictive scaling	Forecasting layer; time-aware validation
Autoscaling and RL	Gandhi et al. [8]; Mao et al. [9]	Scaling as control; learned policies	Simulation-first RL; launch delay matters
Cost-aware provisioning	Chaisiri et al. [10]; Cohen et al. [11]; Cortez et al. [12]	Purchasing-option optimization, chance constraints, production forecasting	Pricing and risk inside the optimizer

The proposed pipeline is summarised in Figure 1.

1. Telemetry and billing data	2. Forecasting	3. SCOS scoring	4. Constrained optimization with safety layer	5. Audit and drift feedback
CPU, memory, requests, latency, cost	Predict demand at horizon h	Rank feasible actions	Choose action or no action	Monitor, retrain, recalibrate

Figure 1. Proposed SCOS pipeline (conceptual design developed in this paper).

8. Discussion: What the Combined Evidence Means

The reviewed studies show that machine learning can forecast cloud demand and support proactive provisioning [5]–[7], [18], but that forecasting is only one step. Accuracy does not guarantee lower spending: a model with lower error can still increase cost if its forecasts trigger unnecessary scaling, ignore data-transfer charges or violate latency targets [24].

Second, cost and risk must be considered together [11]. Reactive thresholds are stable but tend to over-provision [15], while predictive methods are more adaptive but sensitive to model error [26]. The SCOS framework addresses this by scoring savings, performance, utilization, migration cost and risk together, with hard constraints and a rule-based safety layer.

Machine learning should not replace governance. It should provide evidence and recommendations that humans can review, and human approval should remain for high-impact actions such as long-term commitments or provider migration [21]. Simpler, interpretable models should be preferred unless a complex model gives a measurable improvement [28].

Overall, the proposed approach aims to make cloud-cost decisions more predictive, transparent and safe rather than focusing on cost alone.

9. Proposed Experimental Design and Hypotheses

To evaluate the framework, a future study can use public workload traces (for example the Google cluster trace [22], [23] or the Azure public dataset [12]), clearly labelled synthetic data, or organizational telemetry with permission. Public and synthetic data may not reflect real billing structures or operational delays, so they must be identified clearly. The framework will be run in simulation [3] or recommendation mode and compared with the baselines in Section 5.5 over the same chronological test period [25].

Research Hypotheses

H1: Prediction-driven provisioning will reduce total cost compared with CPU-threshold reactive autoscaling at equal SLO compliance [6], [15].

H2: SCOS-based constrained optimization will reduce SLO violations compared with forecast-only provisioning.

H3: Simpler ensemble models will perform within an acceptable margin of LSTM or GRU [28] at lower complexity.

H4: Improvements in forecast accuracy will translate into cost savings less than proportionally [24].

H5: Results will vary by workload type, resource type and cloud provider [13].

Evaluation Metrics

Primary metrics: total cloud expenditure, cost per request, SLO violation rate, resource utilization.

Prediction metrics: MAE, RMSE, MAPE [24] and prediction-interval coverage.

Secondary metrics: over- and under-provisioned capacity, number of scaling actions, scaling delay, failed-action or rollback frequency, and energy or carbon indicators where available [4].

The study should also report sensitivity to the SCOS weights, confidence intervals, effect sizes, missing data and possible negative effects.

10. Limitations and Ethical Considerations

This research has several limitations. Full organizational billing and telemetry data may be restricted by confidentiality. Public datasets may lack pricing, deployment or migration detail [23]. Cloud pricing differs across providers and regions, limiting generalization [13]. A model trained on history may perform poorly during unprecedented events such as sudden traffic surges or major application changes [26]. Finally, an optimizer may recommend actions that are efficient in theory but hard to execute because of dependencies, licensing, compliance or approval processes.

The proposed SCOS framework is still conceptual and has not been experimentally validated. Its weights, features and accuracy require further testing. The reviewed studies also use different datasets, metrics and baselines, so their results cannot be combined directly.

To address these challenges, the framework uses transparent dataset documentation, sensitivity analysis, multiple baselines, drift monitoring [26], a rule-based safety layer and conservative deployment, starting in recommendation mode.

From an ethical and governance perspective, telemetry and billing data must be handled with access control, minimum collection and secure storage, and any real organizational data requires permission. Automated actions should be auditable and reversible, and the framework should not be used to degrade service quality for customers merely to reduce cost.

11. Conclusion

Cloud migration provides scalability and flexibility, but without continuous cost management it can create unnecessary expenditure. The literature shows that machine learning can forecast demand and support proactive provisioning [5]–[7], and also that accurate prediction alone does not prove lower cost [24]. The contribution of this paper is a proposed framework that connects forecasting with a constrained, transparent decision layer, using the Safe Cost Optimization Score to rank feasible actions and a safety layer to protect performance and reliability.

The next research step is empirical validation through a controlled comparison of reactive, forecast-only and SCOS-based strategies using chronological validation [25]. Until such testing is completed, SCOS should be treated as a testable design hypothesis, not a proven solution.

References

[1] M. Armbrust *et al.*, “A view of cloud computing,” *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[2] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, “Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility,” *Future Generation Computer Systems*, vol. 25, no. 6, pp. 599–616, 2009.

[3] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, “CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms,” *Software: Practice and Experience*, vol. 41, no. 1, pp. 23–50, 2011.

[4] A. Beloglazov, J. Abawajy, and R. Buyya, “A taxonomy and survey of energy-efficient data centers and cloud computing systems,” *Advances in Computers*, vol. 82, pp. 47–111, 2011.

[5] S. Di, D. Kondo, and W. Cirne, “Host load prediction in a Google compute cloud with a Bayesian model,” in *Proc. Int. Conf. High Performance Computing, Networking, Storage and Analysis (SC)*, 2012.

[6] Z. Gong, X. Gu, and J. Wilkes, “PRESS: PRedictive Elastic ReSource Scaling for cloud systems,” in *Proc. Int. Conf. Network and Service Management (CNSM)*, 2010.

[7] S. Islam, J. Keung, K. Lee, and A. Liu, “Empirical prediction models for adaptive resource provisioning in the cloud,” *Future Generation Computer Systems*, vol. 28, no. 1, pp. 155–162, 2012.

[8] A. Gandhi, M. Harchol-Balter, R. Raghunathan, and M. A. Kozuch, “AutoScale: Dynamic, robust capacity management for multi-tier data centers,” *ACM Transactions on Computer Systems*, vol. 30, no. 4, 2012.

[9] H. Mao, M. Alizadeh, I. Menache, and S. Kandula, "Resource management with deep reinforcement learning," in *Proc. 15th ACM Workshop on Hot Topics in Networks (HotNets)*, 2016, pp. 50–56.

[10] S. Chaisiri, B.-S. Lee, and D. Niyato, "Optimization of resource provisioning cost in cloud computing," *IEEE Transactions on Services Computing*, vol. 5, no. 2, pp. 164–177, 2012.

[11] M. C. Cohen, P. W. Keller, V. Mirrokni, and M. Zadimoghaddam, "Overcommitment in cloud services: Bin packing with chance constraints," *Management Science*, vol. 65, no. 7, pp. 3255–3271, 2019.

[12] E. Cortez, A. Bonde, A. Muzio, M. Russinovich, M. Fontoura, and R. Bianchini, "Resource Central: Understanding and predicting workloads for improved resource management in large cloud platforms," in *Proc. 26th ACM Symposium on Operating Systems Principles (SOSP)*, 2017, pp. 153–167.

[13] Y. Al-Dhuraibi, F. Paraiso, N. Djarallah, and P. Merle, "Elasticity in cloud computing: State of the art and research challenges," *IEEE Transactions on Services Computing*, vol. 11, no. 2, pp. 430–447, 2018.

[14] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: State-of-the-art and research challenges," *Journal of Internet Services and Applications*, vol. 1, no. 1, pp. 7–18, 2010.

[15] T. Lorido-Botran, J. Miguel-Alonso, and J. A. Lozano, "A review of auto-scaling techniques for elastic applications in cloud environments," *Journal of Grid Computing*, vol. 12, no. 4, pp. 559–592, 2014.

[16] N. Roy, A. Dubey, and A. Gokhale, "Efficient autoscaling in the cloud using predictive models for workload forecasting," in *Proc. IEEE Int. Conf. Cloud Computing (CLOUD)*, 2011, pp. 500–507.

[17] Z. Shen, S. Subbiah, X. Gu, and J. Wilkes, "CloudScale: Elastic resource scaling for multi-tenant cloud systems," in *Proc. ACM Symp. Cloud Computing (SoCC)*, 2011, Art. 5.

[18] R. N. Calheiros, E. Masoumi, R. Ranjan, and R. Buyya, "Workload prediction using ARIMA model and its impact on cloud applications' QoS," *IEEE Transactions on Cloud Computing*, vol. 3, no. 4, pp. 449–458, 2015.

[19] A. Beloglazov and R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers," *Concurrency and Computation: Practice and Experience*, vol. 24, no. 13, pp. 1397–1420, 2012.

[20] T. L. Duc, R. García Leiva, P. Casari, and P.-O. Östberg, "Machine learning methods for reliable resource provisioning in edge-cloud computing: A survey," *ACM Computing Surveys*, vol. 52, no. 5, Art. 94, 2019.

[21] K. Rzadca *et al.*, "Autopilot: Workload autoscaling at Google," in *Proc. 15th European Conf. Computer Systems (EuroSys)*, 2020, Art. 16.

[22] A. Verma *et al.*, "Large-scale cluster management at Google with Borg," in *Proc. 10th European Conf. Computer Systems (EuroSys)*, 2015, Art. 18.

[23] C. Reiss, A. Tumanov, G. R. Ganger, R. H. Katz, and M. A. Kozuch, "Heterogeneity and dynamicity of clouds at scale: Google trace analysis," in *Proc. ACM Symp. Cloud Computing (SoCC)*, 2012, Art. 7.

[24] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *International Journal of Forecasting*, vol. 22, no. 4, pp. 679–688, 2006.

[25] C. Bergmeir and J. M. Benítez, "On the use of cross-validation for time series predictor evaluation," *Information Sciences*, vol. 191, pp. 192–213, 2012.

[26] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Computing Surveys*, vol. 46, no. 4, Art. 44, 2014.

[27] B. Jennings and R. Stadler, "Resource management in clouds: Survey and research challenges," *Journal of Network and Systems Management*, vol. 23, no. 3, pp. 567-619, 2015.