Risk-Aware Contextual Autoscaling using Forecast Credibility and Reinforcement Learning

Amirreza Askarpour, Feb 2026

Problem Statement

Autoscaling Under Uncertainty

Cloud systems must scale resources dynamically to handle workload changes.

Current approaches fall into two categories:

- Reactive autoscaling — scales after workload changes occur
- Forecast-based autoscaling — scales using predicted workload
- Simple Threshold Based scaling (e.g K8s HPA)

However:

- Workloads are uncertain and may drift over time
- Forecasts contain error and uncertainty
- Real systems often have contextual knowledge about future events (campaigns, deployments)

Core issue:

Most autoscalers treat forecasts as deterministic values rather than uncertain beliefs.

Why This Matters

Operational and Economic Impact

- SLA violations degrade performance and user experience
- Under-provisioning leads to latency spikes
- Over-provisioning increases infrastructure cost
- Forecast errors can amplify scaling instability
- Ignoring contextual events reduces proactive capability

Modern cloud environments require decision-making under uncertainty, not simple threshold-based control.

Forecast-Based Proactive Autoscaling

GraphOpticon — Forecast-Driven Scaling [1]

Key characteristics:

- Multi-step workload forecasting
- Global predictive model
- Proactive horizontal scaling
- Evaluated on real cluster traces

Limitations:

- Forecast outputs used directly for scaling decisions
- No explicit probabilistic uncertainty calibration
- No credibility or trust modeling
- No reinforcement learning decision layer
- No contextual event modeling

Forecast quality is assumed, not evaluated or integrated as belief.

Reactive Reinforcement Learning Autoscaling

Q-Learning Based Autoscaling [2]

Key characteristics:

- MDP-based autoscaling formulation
- Q-learning decision mechanism
- Reward includes SLA penalties and resource cost
- Reactive scaling decisions

Limitations:

- No forecasting integration
- No uncertainty representation
- No contextual event awareness
- Learns only from current and past states

Decisions are reactive, without reasoning about uncertain future workload.

Identified Research Gap

Missing Integration in Existing Approaches

Reactive RL

- No future modeling
- No belief representation

Forecast-Based

- Deterministic use of predictions
- No trust estimation

Real-World Requirement

- Uncertain future workload
- Context-aware reasoning
- Explicit risk-aware decision-making

Gap:
There is no unified framework that integrates probabilistic forecasts, forecast credibility scoring, contextual events, and risk-aware reinforcement learning.

Proposed Framework

Risk-Aware Forecast-Augmented RL Architecture

System Components:

- Metrics Collector (current workload(RPS), latency, resource usage)
- Probabilistic Forecast Module
- Forecast Credibility Scoring Module
- Contextual Event Interface
- Risk-Aware Reinforcement Learning Agent
- Cloud Environment (scaling target)

Core idea:

Forecast output is a probability distribution.
Forecast reliability is quantified.
Both are included in the RL state.
The agent learns to scale under uncertainty and contextual signals.

Forecasting Layer

Probabilistic and Event-Conditioned Forecasting

- Multi-horizon workload prediction (e.g Multiple time windows, next 5, 10, 15 minutes)
- Outputs workload quantiles (p10, p50, p90)
- Produces prediction intervals
- Incorporates contextual event inputs
- Continuously monitors calibration quality
- The model outputs a distribution, not a single predicted value.

Contextual Events

Modeling Future Context Information

Structured event definition:

Event =
(type, time window, scope, expected impact, probability, trust score)

Examples:

- Marketing campaign
- CI/CD deployment
- Feature release
- Maintenance window

The agent receives contextual event signals but does not observe the true future workload.

Forecast Credibility Scoring

Dynamic Trust Estimation

Forecast quality is evaluated online using:

- Prediction interval coverage error
- Forecast bias
- Interval width (sharpness)
- Median absolute percentage error

These metrics are combined into a trust score in range [0,1].

This trust score becomes part of the RL state.

The agent learns when to rely on forecasts and when to discount them.

Decision Layer

Risk-Aware Reinforcement Learning

Algorithm:

* Quantile Regression Deep Q-Network (QR-DQN)

Reasons:

- Distributional value estimation
- Supports risk-sensitive decision-making
- Suitable for modeling SLA violation probability

Objective function includes:

- Infrastructure cost
- SLA violation probability
- Scaling oscillation penalty

The agent optimizes long-term performance under uncertainty

Risk-Aware Reinforcement Learning

Decision-Making Beyond Average Performance

Traditional RL Objective:

- Optimize expected (average) reward
- Minimize average cost
- Penalize SLA violations in expectation

Limitation:

Average performance can hide rare but severe SLA violations.

Example:

- Policy A: Low cost, occasional large SLA breaches
- Policy B: Slightly higher cost, stable SLA compliance

Expected-reward optimization may prefer Policy A.

What “Risk-Aware” Means in This Work

The agent considers the distribution of outcomes, not only the mean

- Scaling decisions account for SLA violation probability
- Tail behavior (worst-case performance) influences action selection

We use distributional RL (QR-DQN) to estimate return quantiles rather than a single expected value.

This allows the policy to:

- Reduce probability of SLA breaches
- Balance cost vs. violation risk
- Make more stable scaling decisions under uncertainty

Innovation

Core Innovations of This Work

Forecast Credibility as RL State Feature

- Forecasts are not treated as deterministic values
- Online credibility scoring quantifies trust
- RL agent learns when to rely on predictions

Belief-Based Autoscaling

- Workload forecasts represented as probabilistic quantiles
- Decision-making operates on uncertainty, not point estimates

Risk-Aware Objective

- SLA violation probability modeled explicitly
- Decision policy optimized under risk, not only average reward

Unified Architecture

- Forecasting + credibility scoring + RL integrated into a single decision framework
- Unlike [1] (forecast-only) and [2] (reactive RL), this combines future awareness with uncertainty reasoning

Evaluation Strategy

Experimental Setup

Workload Data:

- Real cluster traces (Google Borg / Alibaba / Azure datasets)
- Historical workload metrics used for forecasting and scaling evaluation

Experimental Design:

- Train forecasting model on historical workload traces
- Compute forecast credibility metrics online
- Train RL agent using forecast quantiles + credibility score
- Compare against baseline approaches

Evaluation Metrics:

- SLA violation rate
- Infrastructure cost
- Resource utilization efficiency
- Scaling stability (oscillation)
- Forecast calibration quality

Evaluation Strategy

Experimental Setup

Baselines:

1. Reactive RL autoscaler [2]
2. Forecast-based proactive autoscaler [1]
3. Proposed risk-aware forecast-augmented RL

All models evaluated under identical workload traces.

Expected Contributions

Research Contributions

- Integration of forecast credibility into RL state
- Structured contextual event modeling for autoscaling
- Explicit risk-aware scaling objective
- Realistic evaluation with contextual event simulation

References

[1]. Song, S., Pan, L., & Liu, S., “A Q-learning based auto-scaling approach for provisioning big data analysis services in cloud environments,” Future Generation Computer Systems, vol. 154, pp. 140–150, 2024, doi:10.1016/j.future.2023.10.003.

[2]. Ahmad, H., Treude, C., Wagner, M., & Szabo, C., “Towards resource-efficient reactive and proactive auto-scaling for microservice architectures,” Journal of Systems and Software, vol. 225, art. no. 112390, 2025, doi:10.1016/j.jss.2025.112390.

[3]. Theodoropoulos, T., Patel, Y. S., Zdun, U., Townend, P., Korontanis, I., Makris, A., & Tserpes, K., “GraphOpticon: A global proactive horizontal autoscaler for improved service performance & resource consumption,” Future Generation Computer Systems, vol. 174, art. no. 107926, 2026, doi:10.1016/j.future.2025.107926.

[4]. Zou, D., Lu, W., Zhu, Z., Lu, X., Zhou, J., Wang, X., Liu, K., Wang, K., Sun, R., & Wang, H., “OptScaler: A collaborative framework for robust autoscaling in the cloud,” Proc. VLDB Endowment, vol. 17, no. 12, pp. 4233–4246, 2024, doi:10.14778/3685800.3685829.

[5]. Pan, Y., Wang, Y., Zhang, Y., Yang, S. B., Cheng, Y., Chen, P., Guo, C., Wen, Q., Tian, X., Dou, Y., Zhou, Z., Yang, C., Zhou, A., & Yang, B., “MagicScaler: Uncertainty-aware predictive autoscaling,” Engineering Applications of Artificial Intelligence, vol. 128, art. no. 107530, 2024, doi:10.1016/j.engappai.2023.107530.

References

[6]. Tutuncuoglu, B. T., “Predictive load resilience: AI-based traffic anticipation and autonomous scaling in web hosting infrastructure,” SSRN Electron. J., Jul. 24, 2025, doi:10.2139/ssrn.5343115.

[7]. Liang, P., Xun, Y., Cai, J., & Yang, H., “Autoscaling of microservice resources based on dense connectivity spatio-temporal GNN and Q-learning,” Future Generation Computer Systems, vol. 174, pp. 140–150, 2026, doi:10.1016/j.future.2025.03.044.

[8]. Li, H., Rao, W., Hu, B., Tian, Y., & Shen, J., “Energy-aware elastic scaling algorithm for microservices in Kubernetes clouds,” Journal of Network and Computer Applications, vol. 242, art. no. 104218, 2025, doi:10.1016/j.jnca.2025.104218.

[9]. Do, T. V., Do, N. H., Rotter, C., Lakshman, T. V., Biro, C., & Bérczes, T., “Properties of horizontal pod autoscaling algorithms and application for scaling cloud-native network functions,” IEEE Trans. Netw. Serv. Manag., vol. 22, no. 2, pp. 1889–1902, 2025, doi:10.1109/TNSM.2025.3532121.

[10]. Chen, L., Jiang, C., Zhong, Q., & Zhang, X., “A deep reinforcement learning approach to cloud resource optimization with response time distributions,” Expert Systems with Applications, vol. 296, art. no. 129081, 2026, doi:10.1016/j.eswa.2025.129081.

[11]. Dogani, J., & Khunjush, F., “Proactive auto-scaling technique for web applications in container-based edge computing using federated learning model,” Journal of Parallel and Distributed Computing, vol. 187, art. no. 104837, 2024, doi:10.1016/j.jpdc.2024.104837.

[12]. Santos, J., Reppas, E., Wauters, T., Volckaert, B., & De Turck, F., “Gwydion: Efficient auto-scaling for complex containerized applications in Kubernetes through reinforcement learning,” Journal of Network and Computer Applications, vol. 234, art. no. 104067, 2025, doi:10.1016/j.jnca.2024.104067.