

INTELLIGENT WORKLOAD PREDICTION AND PROACTIVE AUTO- SCALING FOR COST AND ENERGY OPTIMIZATION IN CLOUD-BASED SYSTEMS

Yashodhan Akhare¹, Dr. Dharendra Kumar Tripathi²

Research Scholar, Mansarovar Global University, Sehore (Bhopal)¹

Research Supervisor, Mansarovar Global University, Sehore (Bhopal)²

ABSTRACT

Dynamic and unpredictable workload behaviour is one of the principal challenges in cloud-resource management. Conventional threshold-based autoscaling mechanisms react only after resource utilization crosses predefined limits, potentially resulting in delayed provisioning, performance degradation, over-provisioning and unnecessary energy consumption. Artificial Intelligence enables cloud systems to move from reactive to proactive resource management by predicting future workload demands and provisioning resources before demand changes occur. This paper proposes an intelligent workload-prediction and proactive autoscaling framework combining historical cloud monitoring data, machine-learning-based forecasting and adaptive resource provisioning. The framework evaluates workload trends and predicts future CPU, memory and service demand before determining virtual-machine or container scaling decisions. Long Short-Term Memory, Transformer-based forecasting and reinforcement-learning approaches are considered for workload prediction and scaling optimization. The model aims to minimize prediction error, response time, SLA violations, resource wastage, cost and energy consumption. Recent research demonstrates that prediction-based resource management is increasingly important for cloud-edge and large-scale distributed environments. The proposed framework contributes to the Ph.D. theme “AI Driven Analysis & Optimization Framework for Cloud Based Systems” by providing a predictive optimization layer capable of supporting intelligent, sustainable and cost-efficient cloud infrastructures.

Keywords: *Cloud computing¹, Workload prediction², Autoscaling³, Artificial intelligence⁴, Machine learning⁵, LSTM⁶, Transformer⁷, Resource provisioning⁸, Energy efficiency⁹*

1. INTRODUCTION

Elasticity is one of the defining characteristics of cloud computing. Cloud resources can theoretically expand or contract according to demand. However, determining when, how much and which resources should be provisioned remains challenging. Consider a cloud-hosted application experiencing a sudden increase from 1,000 to 20,000 user requests. If scaling begins only after CPU utilization exceeds a threshold, additional instances may become available too late. This can result in:

- Increased response time;
- Queue formation;
- Service degradation; and
- SLA violations.

The reverse problem also occurs.

If additional resources remain active after workload demand declines, organizations continue paying for underutilized computing capacity. This creates two major problems:

Under-Provisioning

Allocated resources < Required resources

Result:

- Performance loss;
- Delays;
- SLA violations.

Over-Provisioning

Allocated resources > Required resources

Result:

- Resource wastage;
- Increased energy use;
- Unnecessary cloud cost.

The optimal condition is:

$$\text{Allocated Resources} \approx \text{Actual Demand} \quad \text{Allocated Resources} \approx \text{Actual Demand}$$

Achieving this condition requires an ability to predict future workload. Recent research using 18 months of real production traffic traces demonstrated the importance of workload forecasting for proactive resource allocation in geographically distributed cloud-edge applications. The study integrated statistical and deep-learning forecasting with adaptive autoscaling. AI therefore provides the foundation for converting cloud autoscaling from a reactive to a predictive process.

2. OBJECTIVES

The objectives are:

1. To investigate workload-prediction techniques for cloud environments;
2. To analyse limitations of reactive autoscaling;
3. To develop an AI-based workload forecasting framework;
4. To predict short-term resource requirements;
5. To enable proactive VM/container scaling;
6. To minimize resource over-provisioning and under-provisioning;
7. To reduce cloud operational cost and energy consumption; and
8. To improve SLA compliance and application response time.

3. PROBLEM FORMULATION

Let cloud workload at time t be:

$$W_t$$

The AI model predicts future workload:

$$W_t + k = f(W_t, W_{t-1}, \dots, W_{t-n}) \quad \hat{W}_{t+k} = f(W_t, W_{t-1}, \dots, W_{t-n})$$

where:

- $W_t + k \hat{W}_{t+k}$ = predicted workload;
- k = prediction horizon;
- n = historical observation window.

Resource requirements are then calculated as:

$$R_{t+k} = g(W^t + k)R_{t+k} = g(\hat{W}_{t+k})$$

The objective is to minimize:

$$F = \alpha PE + \beta RT + \gamma C + \delta E + \theta SLAF = \alpha PE + \beta RT + \gamma C + \delta E + \theta SLA$$

where:

- PEPE = prediction error;
- RTRT = response time;
- CC = cost;
- EE = energy consumption;
- SLASLA = SLA violations.

4. Limitations of Traditional Autoscaling

Traditional cloud autoscaling commonly follows rules such as:

- Add VM if CPU > 80%.
- Remove VM if CPU < 30%.
- Although simple, threshold-based rules create several weaknesses.

Reactive Delay

Scaling occurs after workload demand has already changed.

Oscillation

Rapid changes can repeatedly activate and deactivate resources.

Inability to Learn

Fixed thresholds cannot identify recurring patterns.

Multi-Resource Problem

CPU utilization alone may not reflect memory, network or I/O bottlenecks.

Workload Heterogeneity

A threshold appropriate for one application may not be suitable for another. Predictive autoscaling addresses these limitations by estimating future demand before scaling is required.

5. REVIEW OF RECENT LITERATURE

Zhou et al. (2024) Zhou and colleagues demonstrated the increasing importance of DRL in dynamic cloud-resource scheduling. Their review suggests that reinforcement-learning models can continuously adapt resource decisions by observing environmental states and performance outcomes.

Lekkala (2024) Lekkala proposed AI-driven dynamic resource allocation using predictive models and real-time optimization. The study reported that prediction-based allocation could improve utilization while reducing QoS violations relative to traditional mechanisms.

Sanjalawe et al. (2025) Sanjalawe and colleagues reviewed AI-based cloud scheduling and identified dynamic workload behaviour as an important reason for adopting AI-driven resource-management techniques. The study emphasizes the need for systems capable of learning and adapting to heterogeneous workload conditions.

Workload Prediction for Cloud-Edge Applications (2025) A 2025 study proposed a predictive resource-allocation framework based on **18 months of production CDN traffic**. Statistical and deep-learning models were integrated into a feedback-driven autoscaling system. This study provides strong evidence for using predictive workload analytics rather than relying entirely on reactive thresholds.

Manhary et al. (2025) Manhary, Mohamed and Farouk investigated scalable ML-based resource allocation and highlighted limitations associated with reactive DQN approaches during sudden workload spikes. Their work also discussed Transformer-based workload forecasting, demonstrating the emerging integration of prediction and intelligent resource allocation.

Nandagopal et al. (2025) Nandagopal and colleagues developed a hybrid optimization approach integrating task scheduling and Transformer models for energy-efficient cloud computing. The study reflects the emerging trend toward combining predictive AI architectures with metaheuristic resource optimization.

Table 1. Latest Studies Related to Predictive Cloud Optimization

Study	Year	Technique	Research focus
Zhou et al.	2024	DRL	Adaptive resource scheduling
Lekkala	2024	ML/DL prediction	Dynamic allocation
Sanjalawe et al.	2025	AI review	Intelligent scheduling
Cloud-edge workload study	2025	DL forecasting	Proactive autoscaling
Manhary et al.	2025	ML + Transformer	Predictive allocation
Nandagopal et al.	2025	Transformer + optimization	Energy-efficient scheduling

6. RESEARCH GAP

Several gaps remain despite advances in predictive resource management. First, many cloud systems still use reactive threshold-based autoscaling. Second, forecasting and autoscaling are frequently treated as independent problems. Third, prediction accuracy alone does not guarantee optimal resource allocation. Fourth, relatively few approaches simultaneously optimize:

- Response time;
- Cloud cost;
- Energy;
- Utilization; and
- SLA compliance.

Fifth, highly dynamic workload behaviour can reduce the reliability of a forecasting model trained on historical patterns. An integrated system is therefore required in which forecasting, optimization, scaling and continuous feedback operate as a single adaptive framework.

7. Proposed AI-Based Framework

The proposed framework consists of six modules.

Module 1: Workload Monitoring

Collect real-time metrics including:

- Requests per second;
- CPU utilization;
- Memory consumption;
- Network traffic;
- Response time;
- Queue length; and
- Active instances.

Module 2: Historical Data Repository

Time-series data are stored for learning recurring workload patterns.

Module 3: Workload Prediction Engine

Candidate models include:

- ARIMA;
- Random Forest;
- XGBoost;
- LSTM;
- GRU; and
- Transformer.

Module 4: Resource-Demand Estimation

Predicted workload is translated into required computing capacity.

For example:

$$\text{RequiredVM} = \left\lceil \frac{\text{PredictedWorkload}}{\text{CapacityPerVM}} \right\rceil$$

Module 5: Scaling Optimizer

The optimizer determines whether to:

- Scale out;
- Scale in;
- Scale up;
- Scale down; or
- Maintain existing resources.

Module 6: Feedback Learning

Prediction error and observed application performance are used to retrain or adapt the model.

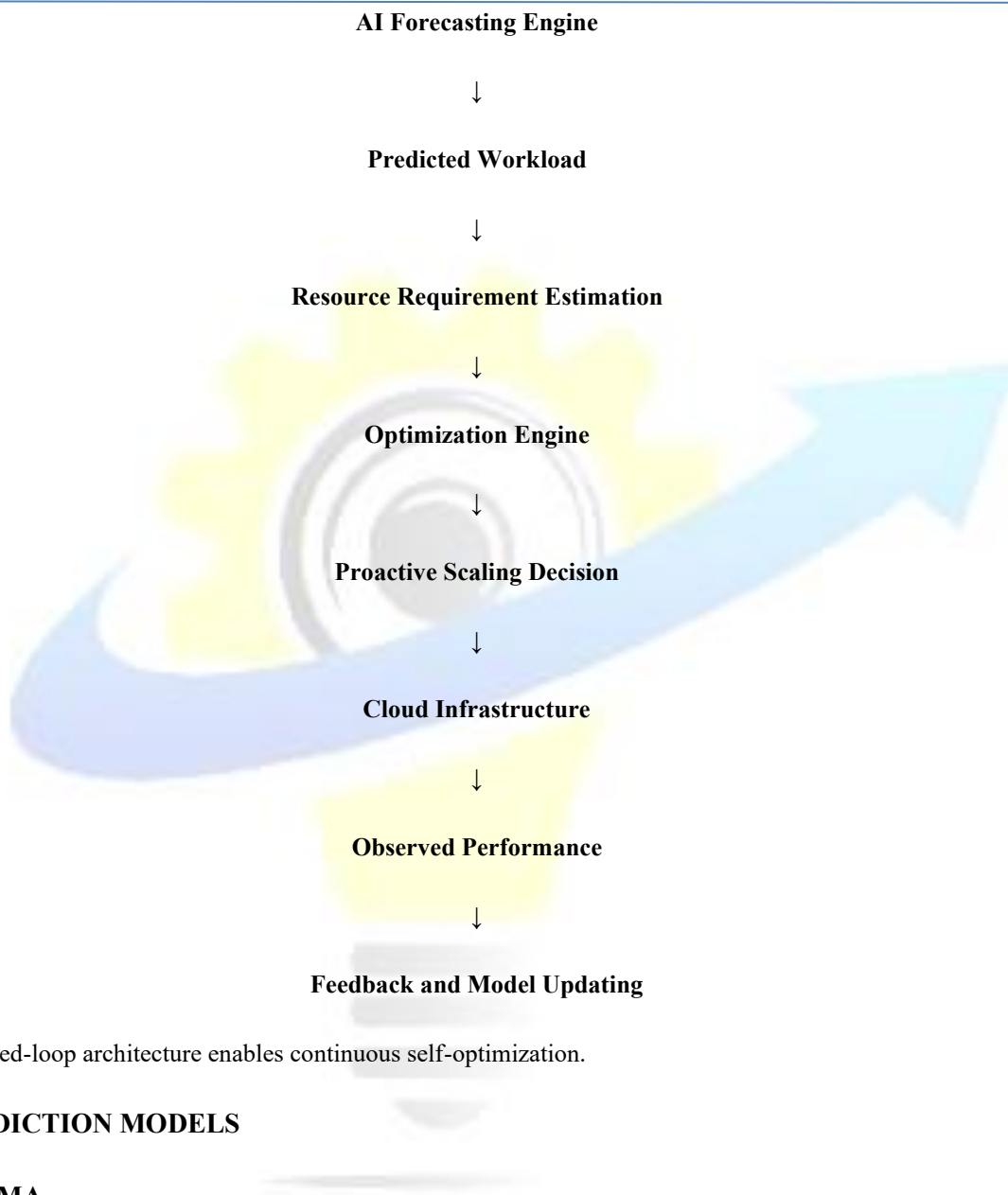
8. PROPOSED SYSTEM FLOW

Historical + Real-Time Workload



Data Pre-processing





This closed-loop architecture enables continuous self-optimization.

9. PREDICTION MODELS

9.1 ARIMA

Useful as a statistical baseline for workloads displaying regular temporal structure.

9.2 Random Forest

Useful for nonlinear relationships involving multiple cloud features.

9.3 LSTM

Long Short-Term Memory networks are particularly suitable for sequential cloud workloads because they retain temporal dependencies.

9.4 GRU

Gated Recurrent Units can provide performance similar to LSTM with reduced model complexity.

9.5 Transformer

Transformer architectures can learn long-range temporal relationships and complex patterns in large-scale workload traces. Recent work increasingly combines Transformer forecasting with cloud-resource optimization, although its computational overhead must be considered for real-time deployment.

10. EXPERIMENTAL METHODOLOGY

Data

Potential datasets include:

- Google Cluster Workload Traces;
- Alibaba Cluster Trace;
- Azure workload traces;
- Bitbrains VM traces; and
- Synthetic application workloads.

Experimental Environments

The framework can be evaluated through:

- CloudSim Plus;
- Kubernetes-based testbed;
- Docker containers;
- AWS/Azure experimental deployment.

Comparison Models

1. Static provisioning
2. Threshold-based autoscaling
3. ARIMA autoscaling
4. Random Forest prediction
5. LSTM prediction
6. Transformer forecasting

7. Proposed AI predictive-optimization framework

11. EVALUATION METRICS

Mean Absolute Error

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

Root Mean Square Error

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

Mean Absolute Percentage Error

$$MAPE = \frac{100}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right|$$

Lower values indicate better workload prediction.

12. RESOURCE OPTIMIZATION METRICS

Resource Utilization

$$RU = \frac{\text{UsedResource}}{\text{ProvisionedResource}} \times 100$$

Cost Saving

$$\begin{aligned} \text{CostSaving} &= \frac{\text{BaselineCost} - \text{ProposedCost}}{\text{BaselineCost}} \times 100 \\ &= \frac{\text{BaselineCost} - \text{ProposedCost}}{\text{BaselineCost}} \times 100 \end{aligned}$$

Energy Saving

$$\begin{aligned} \text{EnergySaving} &= \frac{\text{BaselineEnergy} - \text{ProposedEnergy}}{\text{BaselineEnergy}} \times 100 \\ &= \frac{\text{BaselineEnergy} - \text{ProposedEnergy}}{\text{BaselineEnergy}} \times 100 \end{aligned}$$

SLA Violation

$$SLAV = \frac{\text{RequestsViolatingSLA}}{\text{TotalRequests}} \times 100$$

13. EXPECTED COMPARATIVE PATTERN

Table 2. Expected Performance Characteristics

Method	Prediction	Adaptability	Cost efficiency	SLA performance
Static provisioning	None	Very low	Low	Variable
Threshold scaling	None	Moderate	Moderate	Moderate
ARIMA	Statistical	Moderate	Moderate	Moderate
Random Forest	ML	High	High	High
LSTM	Deep learning	High	High	High
Transformer	Deep learning	Very high	High	Very high
Proposed framework	AI + optimization	Very high	Very high	Very high

14. OPTIMIZATION STRATEGY

A multi-objective optimization score may be used:

$$\begin{aligned}
 \text{Score} &= w_1(1 - PE) + w_2RU - w_3Cost - w_4Energy - w_5SLAScore \\
 &= w_1(1 - PE) + w_2RU - w_3Cost - w_4Energy - w_5SLA
 \end{aligned}$$

The optimization engine selects the scaling action that maximizes this score. This prevents prediction accuracy from becoming the sole objective. A model that predicts workload perfectly but provisions excessive resources would still be inefficient.

15. IMPORTANCE OF PROACTIVE SCALING

Consider a predictable workload spike occurring daily at 10:00 AM.

Reactive Model

- 10:00 AM: Workload increases
- 10:02 AM: CPU crosses threshold
- 10:03 AM: Scaling requested
- 10:05 AM: VM becomes ready

The application remains overloaded for approximately five minutes.

Predictive Model

- 09:55 AM: AI forecasts spike
- 09:56 AM: Scaling requested
- 09:59 AM: VM becomes ready

- 10:00 AM: Increased workload begins

Resources become available before demand increases. This illustrates the practical advantage of prediction-driven scaling.

16. COST-PERFORMANCE TRADE-OFF

Autoscaling is inherently a multi-objective problem. Adding more resources can improve response time but increase cost. Reducing resources can lower cost but increase SLA violations.

Therefore:

Optimal Resources $\neq$ Maximum Resources Optimal Resources $\neq$ Maximum Resources

Instead:

Optimal Resources = Minimum Resources Optimal Resources = Minimum Resources

subject to:

Performance $\geq$ SLA Requirement Performance $\geq$ SLA Requirement

This provides the central optimization principle of the proposed framework.

17. ENERGY-EFFICIENT CLOUD OPTIMIZATION

Cloud-resource over-provisioning also has environmental implications. Idle servers continue consuming electricity even when utilization is low. Predictive consolidation can enable:

- Shutting down unnecessary VMs;
- Placing idle servers into low-power states;
- Consolidating tasks;
- Minimizing resource fragmentation; and
- Reducing data-centre energy consumption.

Recent 2025 research has increasingly connected AI-based task scheduling with explicit energy-efficiency objectives, including hybrid Transformer and optimization architectures.

18. CONCLUSION

Cloud workloads are dynamic, heterogeneous and increasingly difficult to manage using conventional reactive resource-allocation techniques. Threshold-based autoscaling cannot anticipate future resource demand and may result in delayed provisioning during sudden workload increases or unnecessary resource consumption during

periods of low demand. The present paper proposes an AI-driven predictive autoscaling framework integrating workload monitoring, deep-learning-based forecasting, resource-demand estimation and multi-objective optimization. LSTM, GRU and Transformer models provide suitable candidates for workload forecasting, while the optimization layer converts predicted demand into proactive scaling decisions.

The proposed framework aims to minimize prediction error, response time, cloud cost, energy consumption and SLA violations while maximizing resource utilization. The principal contribution lies in integrating prediction and optimization within a continuous feedback loop.

19. REFERENCES

- [1] Zhou, G., Tian, W., Buyya, R., et al. (2024). Deep reinforcement learning-based methods for resource scheduling in cloud computing: A review and future directions. *Artificial Intelligence Review*, 57, 124.
- [2] Lekkala, C. (2024). AI-driven dynamic resource allocation in cloud computing: Predictive models and real-time optimization. *Journal of Artificial Intelligence, Machine Learning and Data Science*, 2(2).
- [3] Sanjalawe, Y., Al-E'mari, S., Fraihat, S., et al. (2025). AI-driven job scheduling in cloud computing: A comprehensive review. *Artificial Intelligence Review*, 58, 197.
- [4] Nandagopal, M., Manavalan, T., Kumar, K. P., et al. (2025). Enhancing energy efficiency in cloud computing through task scheduling with hybrid cuckoo search and transformer models. *Discover Computing*, 28, 199.
- [5] Manhary, F. N., Mohamed, M. H., & Farouk, M. (2025). A scalable machine learning strategy for resource allocation in database. *Scientific Reports*, 15, 30567.
- [6] Workload Prediction for Proactive Resource Allocation in Large-Scale Cloud-Edge Applications. (2025). *Electronics*, 14(16), 3333.