

# Development of an AI-Based Cloud Auto-Scaling System for Predicting and Managing Resource Demand.

Deane Brickxylle T. Sanipa, Janred Saito, Jaspreet Singh, Daron Gerard Morton

<sup>1</sup> College of Computing and Information Sciences, System Plus College Foundation, Angeles City, Philippines

Corresponding Author Email: [brickxyllesanipa@gmail.com](mailto:brickxyllesanipa@gmail.com), [saitojandred@gmail.com](mailto:saitojandred@gmail.com), [jashan5star@gmail.com](mailto:jashan5star@gmail.com), [dgmnozar@gmail.com](mailto:dgmnozar@gmail.com)

## ABSTRACT

Cloud applications experience changing workloads, making it necessary to adjust computing resources according to demand. Traditional auto-scaling commonly reacts to current resource usage, which may lead to delayed scaling, over-provisioning, or under-provisioning. This study proposes the development of an AI-based cloud auto-scaling system that predicts future resource demand and automatically manages cloud resources. The proposed system will use historical and real-time CPU and memory data to train an AI model for predicting future resource requirements. Its main components will include resource monitoring, data processing, AI-based prediction, scaling decisions, automatic resource scaling, and performance monitoring. The study will use a Descriptive-Developmental Research Design with a Quantitative Approach and an Agile-inspired iterative development methodology. The system will be tested through Unit Testing, Integration Testing, and User Acceptance Testing. Evaluation will consider prediction accuracy, resource utilization, scaling performance, and application performance under different workload conditions. The study aims to combine monitoring, AI prediction, and automatic scaling in one practical system that can help reduce unnecessary resource allocation while maintaining sufficient resources for changing demand.

**Keywords:** cloud computing, auto-scaling, artificial intelligence, machine learning, workload prediction, resource management

## SECTION I — INTRODUCTION (THE PROBLEM AND ITS SETTING)

### 1.1 Introduction

- Most apps and websites we use today run on the cloud. This means the owner rents servers instead of keeping their own machines. When a lot of users show up, the app needs more computing power, and when only a few are around, it needs less. Adding or removing that power is called scaling.

Getting the timing right is the hard part. A lot of systems just react. They add power once the app is already slow, or they leave too much running all day and pay for nothing. Newer studies use AI to guess the demand first so the system can get ready early (Xue et al., 2022; Suleiman et al., 2024). Still, most of them focus on testing one prediction method. Only a few put monitoring, prediction, and scaling together in one system that actually runs.

That is what this study will build. The system watches CPU and memory use, predicts what is coming next, and scales the cloud resources up or down on its own. The study tries to answer these questions:

1. How accurately can an AI model predict CPU and memory demand?
2. How can the predictions be used to scale cloud resources automatically?

### 3. How much can the system reduce over-provisioning and under-provisioning?

#### 1.2 Background of the Study

- Businesses, schools, and online services all use the cloud, and the number of people using them changes throughout the day. Basic auto-scaling usually runs on rules, like "add a server when CPU goes above 80%." It's simple, but it only reacts. New resources take time to start, so users may already feel the slowdown before the help arrives.

That's why researchers turned to predictive scaling. Xue et al. (2022) used meta reinforcement learning to do it. Chouliaras and Sotiriadis (2022, 2023) built scaling frameworks based on workload and on adapting to change. Suleiman et al. (2024) used LSTM models to predict several steps ahead. Arya et al. (2024) made a scaling system that relies on workload prediction, while Nashaat et al. (2026) used a model that keeps adjusting as workloads shift. Two reviews, Khan et al. (2022) and Yekta and Shahhoseini (2023), show how common machine learning has become in this area.

There are still problems to sort out. A wrong prediction can leave the system with too few resources or too many. Workloads keep changing, so models need to keep learning from new data (Hao et al., 2023). Complex models also eat a lot of computing power, which is hard for anyone with limited resources. And many studies are tested on just one dataset or platform, so it's unclear how well they work in real use. This study tries to help by putting monitoring, prediction, and scaling into one system that can be built and tested.

#### 1.3 Objectives of the Study

- **General Objective:** To develop an AI-based cloud auto-scaling system that predicts future resource demand and manages cloud resources automatically.
- **Specific Objectives:**
  - 1.To develop an AI model that uses past and current CPU and memory data to predict future resource needs.
  - 2.To build an auto-scaling function that adds or removes cloud resources based on what the model predicts.
  - 3.To combine real-time monitoring, prediction, and scaling in one system.
  - 4.To reduce over-provisioning and under-provisioning.
  - 5.To test the system in a cloud test environment by checking prediction accuracy, resource usage, and application performance.

#### 1.4 Scope and Limitation of the Study

- **Scope:**
  - The system collects CPU and memory data from a test application running in the cloud.
  - The model is trained and tested using [name of public dataset] and the data from the test application.
  - The system is built and tested in [your cloud or virtual machine setup].
  - Traffic is simulated with [Locust / JMeter / k6], covering normal use, sudden spikes, and slow periods.
  - The system predicts demand and scales resources up or down automatically.
- **Limitation:**
  - The system is not deployed in a real school and does not use multi-year school data, so it can't capture the yearly usage patterns of any one organization.
  - Results may not hold for other cloud platforms, applications, or workloads.
  - Only CPU and memory are covered. [Storage and network are not included.]
  - Prediction accuracy depends on how good the data is.

### 1.5 Significance of the Study

- The people who manage cloud systems will probably feel the biggest change. Instead of watching usage all day and adjusting things by hand, they can let the system handle most of it and focus on other work. Organizations that run apps in the cloud can stop paying for servers that sit idle, and they still get enough power when traffic goes up. A school with an online portal, for example, could use this kind of system in the future when enrollment or exam season brings a rush of users. For the people using those apps, the benefit is simple: fewer slow pages and fewer times when the service just doesn't load. Future researchers can also use this study as a starting point for systems that join monitoring, prediction, and scaling. As for us, the researchers, building it gives us real practice in cloud computing and machine learning.

### 1.6 Definition of Terms

- **Cloud computing** means using servers, storage, and other computing resources over the internet instead of owning them. In this study, it is the environment where the system is built and tested.

**Auto-scaling** is when a system adds or removes computing resources by itself, depending on how much is needed. Here, it refers to the part of the system that changes the cloud resources so nobody has to do it manually.

**Predictive auto-scaling** does the scaling before demand changes, using a forecast, instead of after. In this study, the scaling decision comes from what the model predicts.

**Machine learning** is a way for computers to learn patterns from data and use them to predict what comes next. In this study, it is the model trained on usage data to forecast future demand.

**Workload** is how much work a system has to handle at one time. For this study, it is the demand placed on the test application, measured through CPU and memory use.

**Public workload dataset** is a free, shared record of how real cloud servers were used over time. The one used here is [name of dataset], which trains and tests the model.

**Over-provisioning** is having more resources than the app needs. In this study, it means any time the allocated resources are well above the actual demand.

**Under-provisioning** is the opposite, having too few resources for the demand. Here, it means the allocated resources can't keep up with the actual demand, and the app may slow down.

## SECTION II — LITERATURE REVIEW

### 2.1 Introduction to the Review

- Cloud computing is widely used by websites and applications today. Since the number of users can change, cloud systems need to adjust their resources. Auto-scaling helps by adding or removing resources when needed. However, traditional auto-scaling usually reacts only after resource usage increases. Because of this, researchers have started using Artificial Intelligence (AI) and Machine Learning (ML) to predict future resource demand. This chapter discusses studies about cloud computing, auto-scaling, workload prediction, and the use of AI in managing cloud resources.

## 2.2 Review of Related Literature and Studies

- Cloud computing allows applications to use computing resources through the internet. These resources can be increased or decreased depending on the number of users. Auto-scaling is a cloud feature that automatically adds or removes resources. For example, more resources can be added when CPU usage becomes high and removed when usage becomes low. Machine learning can help improve auto-scaling by predicting future resource needs. Xue et al. (2022) studied the use of machine learning for cloud resource management. Suleiman et al. (2024) used an LSTM model to predict future workload. Other studies by Chouliaras and Sotiriadis (2022, 2023) and Arya et al. (2024) also explored workload prediction and cloud scaling. These studies show that AI can help cloud systems prepare for future changes in workload. However, prediction is not always perfect. Wrong predictions can cause too many or too few resources to be used. Hao et al. (2023) and Nashaat et al. (2026) also discussed the importance of adapting to changing workloads. This is important because cloud usage can change over time. Overall, the studies show that AI and machine learning can help improve cloud resource management and auto-scaling.

## 2.3 Conceptual Framework

- The study uses the **Input-Process-Output (IPO) model**.

### Input

- CPU usage data
- Memory usage data
- Historical workload data • Real-time workload data
- Cloud resources

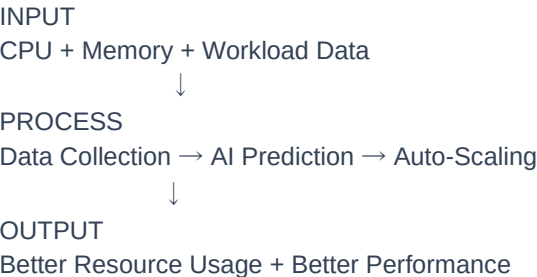
### Process

1. Collect data.
2. Prepare the data.
3. Train the AI model.
4. Predict future resource demand.
5. Decide whether to add or remove resources.
6. Automatically scale the cloud resources.
7. Monitor the system performance.

### Output

- Predicted resource demand
- Automatic resource scaling
- Better resource usage • Less over-provisioning
- Less under-provisioning
- Better application performance

### Simple flow:



## 2.4 Synthesis

- The studies reviewed show that traditional auto-scaling mainly reacts to current resource usage. AI-based predictive scaling can help by predicting future demand before it happens. Previous studies have shown that machine learning can be used to predict workload and improve cloud resource management. However, there are still challenges such as inaccurate predictions and changing workloads. This study will combine monitoring, AI prediction, and automatic scaling into one system. The goal is to predict future CPU and memory demand and automatically adjust cloud resources to improve performance and reduce unnecessary resource usage.

## SECTION III — METHODOLOGY

### 3.1 Research Design

- This study will use a **Descriptive-Developmental Research Design with a Quantitative Approach**. The developmental aspect will focus on creating and testing the AI-based cloud auto-scaling system, while the descriptive-quantitative aspect will measure its performance under different workload conditions. The study will examine prediction accuracy, resource utilization, scaling behavior, and application performance. The system will use historical and real-time CPU and memory data to predict future resource demand. The prediction will then be used to determine whether cloud resources should be increased or decreased.

### 3.2 System Development Methodology

- The study will use an **Agile-inspired iterative methodology**. Development will be performed through planning, data preparation, AI model development, system development, integration, and testing. Each component will be tested before being integrated into the complete system.

**Planning:** Identify system requirements, data, environment, and evaluation measures.

**Data Preparation:** Collect and prepare historical and real-time workload data.

**AI Development:** Train and test the prediction model.

**System Development:** Develop monitoring, prediction, and auto-scaling components.

**Integration and Testing:** Connect the components and evaluate the complete system. Selected evaluators will be involved in system evaluation. The exact number and profile of evaluators will be finalized during the evaluation phase.

### 3.3 Requirement Specification and Analysis

- The system requirements are based on the study objectives and conceptual framework. The main functional requirements are resource monitoring, workload data collection, data preparation, resource demand prediction, scaling decisions, automatic resource scaling, performance monitoring, and data recording. **Non-functional requirements** include performance, reliability, usability, maintainability, scalability, and security. The system analysis will be represented using a **Data Flow Diagram (DFD)** showing the movement of workload data from monitoring to prediction and scaling, and an **Entity-Relationship Diagram (ERD)** showing the relationships among monitoring, prediction, scaling-action, and system-performance data.

### 3.4 System Design Specification

- **Logical Specification:** The system will consist of a Resource Monitoring Module, Data Processing Module, AI Prediction Module, Scaling Decision Module, Auto-Scaling Module, and Monitoring/Reporting Module. The general flow is: Monitor → Process → Predict → Decide → Scale → Monitor.

- **Physical Specification:** The system will require a computer or laptop, internet connection, cloud or virtual-machine resources, and appropriate software for programming, machine learning, data storage, and workload generation. The exact cloud environment and software stack will be finalized during implementation.

### 3.5 Implementation

- Implementation will transform the system design into a working prototype. Historical and real-time CPU and memory data will be collected and prepared for the AI model. The prediction model will be trained and tested, while the monitoring and auto-scaling components will be developed separately. After individual components are tested, they will be integrated into one system. The completed system will continuously follow the cycle: **Monitor → Predict → Decide → Scale → Monitor Again.**

### 3.6 Testing and Evaluation

- The system will undergo **Unit Testing, Integration Testing, and User Acceptance Testing (UAT)**. Unit testing will verify individual components, while integration testing will verify communication among monitoring, prediction, scaling, and performance modules. UAT will assess functionality, usability, responsiveness, clarity, and reliability. System performance will be evaluated using **prediction accuracy, resource utilization, application performance, and scaling performance**. Prediction accuracy may be measured using MAE, RMSE, and MAPE where appropriate. Testing will include normal workload, sudden workload increases, low workload, and changing workload conditions. The testing environment will use a cloud or virtual-machine setup. A workload-generation tool such as Locust, JMeter, or k6 may be used, with the final tool to be selected during implementation.

## REFERENCES

### Reference Guidelines

- Arya, T. F., Rachmadi, R. F., & Affandi, A. (2024). Cloud node auto-scaling system automation based on computing workload prediction. *Jurnal RESTI*, 8(5), 597–606. <https://doi.org/10.29207/resti.v8i5.5928>
- Chouliaras, S., & Sotiriadis, S. (2022). Auto-scaling containerized cloud applications: A workload-driven approach. *Simulation Modelling Practice and Theory*, 121, 102654. <https://doi.org/10.1016/j.simpat.2022.102654>
- Chouliaras, S., & Sotiriadis, S. (2023). An adaptive auto-scaling framework for cloud resource provisioning. *Future Generation Computer Systems*, 148, 173–183. <https://doi.org/10.1016/j.future.2023.05.017>
- Hao, H., Chu, Z., Zhu, S., Jiang, G., Wang, Y., Jiang, C., Zhang, J., Jiang, W., Xue, S., & Zhou, J. (2023). Continual learning in predictive autoscaling. *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 4616–4622. <https://doi.org/10.1145/3583780.3615463>
- Khan, T., Tian, W., Zhou, G., Ilager, S., Gong, M., & Buyya, R. (2022). Machine learning (ML)-centric resource management in cloud computing: A review and future directions. *Journal of Network and Computer Applications*, 204, 103405. <https://doi.org/10.1016/j.jnca.2022.103405>
- Nashaat, M., Moussa, W., Rizk, R., & Saber, W. (2026). Dynamic machine learning approach for workload prediction in cloud environments. *Scientific Reports*, 16, 10983. <https://doi.org/10.1038/s41598-026-40777-z>

Rossi, A., Visentin, A., Carraro, D., Prestwich, S., & Brown, K. N. (2025). Forecasting workload in cloud computing: Towards uncertainty-aware predictions and transfer learning. *Cluster Computing*, 28, 258. <https://doi.org/10.1007/s10586-024-04933-2>

Suleiman, B., Alibasa, M. J., Chang, Y.-Y., & Anaissi, A. (2024). Predictive auto-scaling: LSTM-based multi-step cloud workload prediction. In *Service-oriented computing – ICSOC 2023 workshops* (pp. 5–16). Springer. [https://doi.org/10.1007/978-981-97-0989-2\\_1](https://doi.org/10.1007/978-981-97-0989-2_1)

Xue, S., Qu, C., Shi, X., Liao, C., Zhu, S., Tan, X., Ma, L., Wang, S., Wang, S., Hu, Y., Lei, L., Zheng, Y., Li, J., & Zhang, J. (2022). A meta reinforcement learning approach for predictive autoscaling in the cloud. *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 4290–4299. <https://doi.org/10.1145/3534678.3539063>

Yekta, M., & Shahhoseini, H. S. (2023). A review on machine learning methods for workload prediction in cloud computing. *2023 13th International Conference on Computer and Knowledge Engineering (ICCKE)*. <https://doi.org/10.1109/ICCKE60553.2023.10326297>