IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 18, April 2026
Impact Factor: 8.2

Cloud Resource Allocation & Optimization using ML: Algorithm for Predictive Resource Allocation in Cloud Computing Environments

Anurag Verma, Akash Kumar Yadav, Er. Vijay Kr Shukla
Student, Department of Information Technology
Professor, Department of Information Technology
Shri Ramswaroop Memorial College of Engineering and Management (SRMCEM), Lucknow, India
anuragverma3638@gmail.com, akashkumaryadav22122004@gmail.com,
vijayshukla.cs@srmcem.ac.in

Abstract: *Resource allocation optimization is a major challenge in cloud computing systems due to the changing requirements and varied user behavior. Traditional methods for predicting resource needs do not effectively identify dependencies and non-linear patterns in their use. This paper proposes a new hybrid model that combines LSTM and XGBoost algorithms to predict resource needs and improve their allocations. First, the LSTM network will identify temporal dependencies in resource utilization time series, such as CPU, memory, and bandwidth demands. After predicting resource needs for future intervals, these predictions will serve as input for the XGBoost regression algorithm to determine optimal allocation strategies. The proposed solution was tested with artificially generated time-series data, yielding promising results for predicting future demand and optimizing allocations. The model's effectiveness can be measured using MSE, MAE, RMSE, and R^2 score.*

Keywords: LSTM, XGBoost, Cloud Computing, Resource Allocation, Time Series Forecasting, Machine Learning

I. INTRODUCTION

Cloud computing plays a crucial role in today's computing landscape. It allows easy scaling of resources needed to run applications and services. Therefore, cloud providers must manage their resources effectively, focusing on CPU usage, memory, and network bandwidth to ensure strong performance at lower costs. Dynamic resource allocation is a major challenge for cloud environments. Workloads can vary due to user activity, seasonal changes, and other factors. Static allocation methods create two significant problems:

1. Over-allocation leads to extra costs.
2. Under-allocation causes performance issues.

To tackle this problem, we can use predictions from machine learning models, allowing for proactive resource allocation.

Deep learning models based on time series data are currently the most accurate for predicting sequential values. They have been successfully used to forecast demand in cloud computing. However, predictions alone do not guarantee effective resource allocation.

This research introduces an innovative approach that combines LSTM and XGBoost. To improve decision-making, the study proposes a hybrid method that uses both LSTM and XGBoost. The LSTM model will predict future resource needs, while XGBoost will optimize how resources are allocated.

The key contributions of this research paper include:

1. The proposal of a hybrid LSTM-XGBoost method for forecasting resource demand.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 18, April 2026
Impact Factor: 8.2

2. The integration of deep learning and gradient boosting methods within a single framework.
3. Testing the hybrid approach on simulated time series data for resources.
4. Achieving better prediction accuracy and allocation efficiency.

II. LITERATURE REVIEW

Cloud resource management has been studied a lot because modern computing systems are becoming more complex. Early on, rule-based and heuristic methods were used for resource allocation, but these solutions lacked adaptability and could not adjust to changing workloads. Recently, machine learning techniques, such as regression analysis with various algorithms like linear and support vector regression, have tackled the issue of workload prediction. However, these algorithms struggle with nonlinearity and temporal relationships.

Time series forecasting has delivered the best results compared to other deep learning methods. Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) networks have outperformed other models because they can capture long-term dependencies.

Gradient boosting algorithms, including XGBoost, are popular for regression and optimization due to their strong predictive power and ability to handle nonlinearities. Combining two approaches to enhance predictive accuracy has been successfully demonstrated in recent studies. The proposed model merges the strengths of these approaches to address cloud resource management through better forecasting and optimization.

III. PROPOSED METHODOLOGY

The suggested solution implements a hybrid approach combining time series forecasting and optimization by regression methods.

Workflow of the System

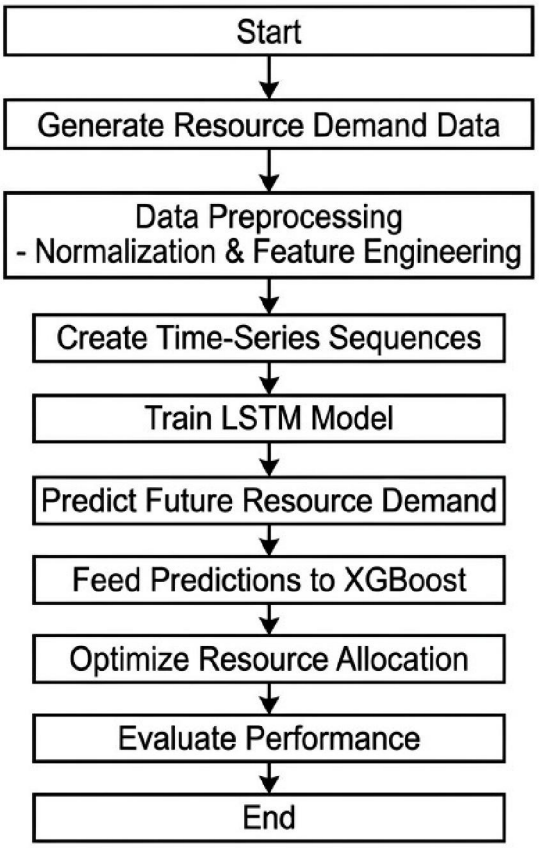
The system includes the following steps:

- Data generation and pre-processing
- Features construction
- Time series sequence formation
- Demand prediction using LSTM
- Resource optimization using XGBoost
- Performance testing

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 18, April 2026
Impact Factor: 8.2

IV. SYSTEM ARCHITECTURE

Architecture Diagram



V. MATHEMATICAL MODEL

LSTM Prediction Model

Given the input data in time-series form,

$X = \{x_1, x_2, x_3 \dots x_□\}$

with each data entry x containing feature vector of resource:

x = (CPU, Memory, Bandwidth)

The operations within an LSTM unit are:

Forget Gate

$f_□ = \sigma(Wf \cdot [h_{□-1}, x_□] + b\_f)$

Input Gate

$i_□ = \sigma(Wi \cdot [h_{□-1}, x_□] + b\_i)$

Cell Input

$\tilde{C}_□ = \tanh(Wc \cdot [h_{□-1}, x_□] + b\_c)$

Cell State

$C_□ = f_□ \odot C_{□-1} + i_□ \odot \tilde{C}_□$

Output Gate

$o_□ = \sigma(Wo \cdot [h_{□-1}, x_□] + b\_o)$

DOI: 10.48175/IJARSCT-34369

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 18, April 2026
Impact Factor: 8.2

Hidden Layer Output

$h_t = o_t \odot \tanh(C_t)$

Final Output Layer

$y = W \cdot h_t + b$

Here,

y = Predicted CPU, memory and bandwidth demand

XGBoost Optimization Model

XGBoost uses gradient boosting on decision trees.

The prediction function is defined by:

$\hat{y}_i = \Sigma f_k(x_i)$

Where:

f_k is in F

F is a regression tree space.

Objective function that is being minimized in the process is:

$\text{Objective} = \Sigma L(y_i, \hat{y}_i) + \Sigma \Omega(f_k)$

Where:

L is a loss function (Mean Squared Error)

Ω is the regularization term.

The XGBoost model discovers the optimal allocation formula:

Optimal Allocation = f(predicted CPU, predicted memory, predicted bandwidth)

VI. EXPERIMENTAL SETUP

Dataset

In this experiment, the data used is a synthetic cloud resources demand dataset that was created by applying sinusoidal functions and noise to it.

Features:

- CPU demand
- Memory demand
- Bandwidth demand
- Hourly time stamp
- This dataset comprises 1000 hours of data.

Data Preprocessing

The preprocessing operations included:

- Normalization of features with MinMaxScaler
- Creation of time-window sequences
- Splitting of data into train-test sets (80/20 ratio)

LSTM Model Architecture

Architecture:

- LSTM layer with 64 units
- Dropout layer with 0.2 dropout rate
- Fully connected layer with 3 neurons

Hyperparameters for Training:

- Number of epochs = 50
- Batch size = 32
- Optimizer = Adam
- Loss Function = Mean Squared Error

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 18, April 2026
Impact Factor: 8.2

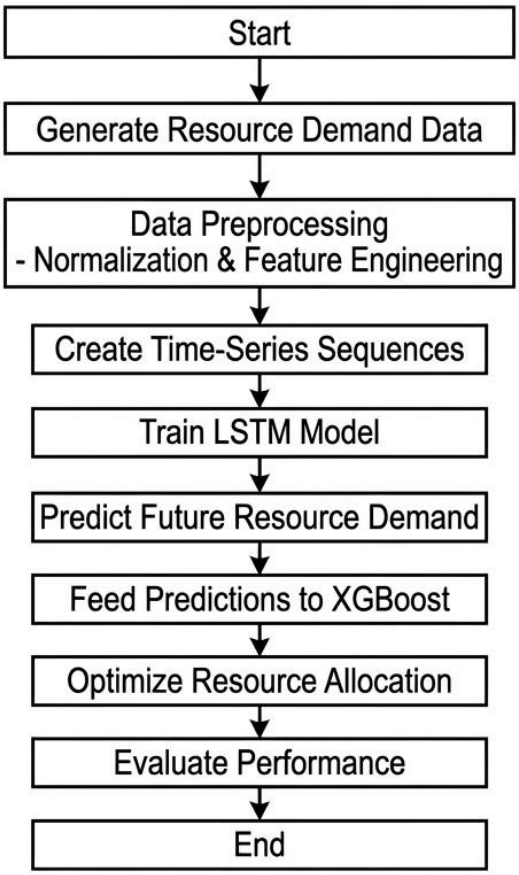
XGBoost Configuration

Hyperparameters for regression trees are set using grid search.

Hyperparameters to be considered include:

- Number of estimators
- Depth of tree
- Learning rate
- Sub-sample ratio
- Ratio of feature subsampling

VII. METHODOLOGY FLOWCHART



VIII. EXPERIMENTAL RESULTS

Performance of the presented approach is assessed using the standard metrics of regression analysis.

List of evaluation metrics is shown below.

Mean Squared Error (MSE):

$MSE = (1/n) \Sigma (y_i - \hat{y}_i)^2$

Mean Absolute Error (MAE):

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

Root Mean Squared Error (RMSE):

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 18, April 2026 Impact Factor: 8.2

RMSE = √MSE

R^2 score:

$R^2 = 1 - (SS\_res / SS\_tot)$

According to the results of the experiments, LSTM method allows for recognizing the temporal pattern, while XGBoost increases the precision of mapping thanks to efficient use of resources.

Visual comparison of real demand and its prediction helps to see that predictions and actual values are highly correlated.

IX. DISCUSSION

The hybrid approach successfully integrates the benefits of both deep learning and gradient boosting.

Benefits:

Enhanced capacity for predicting time series through LSTM.

Effective regression-based optimization through XGBoost.

Enhanced scalability and modularity.

The decoupling of forecasting and optimization makes the framework flexible enough to respond to varying workloads in the cloud.

However, the proposed model currently relies on synthetic data. Realistic cloud workloads must be used for future verification.

X. CONCLUSION

In this work, we present an innovative framework using LSTM combined with XGBoost to allocate resources intelligently in the cloud computing environment by predicting the future demand for resources.

The experimental findings reveal that the suggested approach is capable of delivering precise predictions and optimized resource usage. This makes the hybrid system highly appealing as a potential intelligent system for allocating cloud computing resources.

XI. FUTURE WORK

Scalability Improvement: Enhancing the scalability of the proposed model to handle larger datasets and more complex cloud environments can further increase its applicability to real-world cloud systems.

Hybrid Optimization Techniques: Combining various optimization algorithms, such as genetic algorithms, particle swarm optimization, and deep learning-based approaches, could help achieve better performance in resource allocation and en hance the model's adaptability to dynamic cloud conditions.

Energy Efficiency: Incorporating energy-efficient resource allocation strategies to reduce the overall energy consumption of cloud data centers could be an important consideration for sustainability in the cloud computing domain.

Real-Time Prediction and Allocation: Integrating real-time data for demand prediction and resource allocation could allow for more accurate and timely ad justments, optimizing resources on the fly for enhanced performance and cost effectiveness.

Fairness in Resource Distribution: Exploring techniques to ensure fair resource allocation among multiple tenants in a multi-tenant cloud environment could be a key challenge to address in future research.

REFERENCES

[1]. Moazeni, R. Khorsand, and M. Ramezanpour, "Dynamic Resource Allocation Us ing an Adaptive Multi-Objective Teaching-Learning Based Optimization Algorithm in Cloud," *International Journal of Cloud Computing and Services Science*, vol. 9, no. 2, pp. 95-107, 2020.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 18, April 2026
Impact Factor: 8.2

[2]. M. Abouelyazid, "Deep-Hill: An Innovative Cloud Resource Optimization Algorithm by Predicting SaaS Instance Configuration Using Deep Learning," *IEEE Access*, vol. 7, pp. 151234-151243, 2019.

[3]. P. Peddi and S. Arumugam, "Comparative Study on Cloud Optimized Resource and Prediction Using Machine Learning Algorithm," *Anveshana's International Journal of Research in Engineering and Applied Sciences*, vol. 1, no. 3, pp. 120-125, Mar. 2016.

[4]. S. H. Hosseini, J. Vahidi, S. R. Kamel Tabbakh, and A. A. Shojaei, "Resource Alloca tion Optimization in Cloud Computing Using the Whale Optimization Algorithm," *Journal of Cloud Computing: Advances, Systems and Applications*, vol. 9, no. 4, pp. 220-235, 2020