Journal of Information & Optimization Sciences
ISSN 0252-2667 (Print), ISSN 2169-0103 (Online)
Vol. 47 (2026), No. 5-A, pp. 1791–1798
DOI : 10.47974/JIOS-2270

Implementing machine learning for optimized resource allocation in cloud computing environments

Leena Bharat Chaudhari [‡]
Department of Electronics and Telecommunication
Bharati Vidyapeeth College of Engineering
Lavale
Pune 412115
Maharashtra
India

Manisha Sagar Pawar [†]
Department of Engineering Science and Humanities
Vishwakarma Institute of Technology
Pune 411037
Maharashtra
India

Anjali Bhardwaj [§]
Department of Computer Science & Engineering
Noida International University
Greater Noida 203201
Uttar Pradesh
India

Mahesh Soni *
Nassau Financial Group
Hartford 06102
Connecticut
United States of America

[‡] *E-mail:* leenapc23@gmail.com
[†] *E-mail:* manisha.pawar@vit.edu
[§] *E-mail:* anjali.bhardwaj@niu.edu.in
* *E-mail:* mahesh.times@gmail.com (Corresponding Author)

Deepak Suresh Asudani [@]
Department of Computer Science and Engineering
Symbiosis Institute of Technology
Nagpur Campus
Symbiosis International (Deemed University)
Nagpur 440008
Maharashtra
India

C. K. Rajashri [#]
Department of Computer Science
Meenakshi College of Arts and Science
Meenakshi Academy of Higher Education and Research
Chennai 600078
Tamil Nadu
India

Abstract

Self-driving cars need to learn how to drive in places that change all the time, like when the weather changes, the traffic patterns change, or the noise from sensors changes. RDS makes guidance more reliable and useful, which helps these cars get ready for and deal with these kinds of unknowns. These methods help self-driving cars get ready for problems, change their routes on the fly, and keep working well even when things go wrong. Compared to traditional methods, the results show improvements of up to 25% in utilisation, ~70% reduction in SLA violations, 23% lower energy consumption, and 24% lower cost.

Subject Classification: *46N10, 37N40.*

Keywords: *Cloud computing, Resource allocation, Machine learning, LSTM, Reinforcement learning, Hybrid framework, SLA compliance, Energy efficiency, Cost optimization.*

1. Introduction

One of the biggest IT changes is cloud computing. It lets you pay for computer storage, networking, and processing power on demand. The rapid adoption of cloud services in healthcare, finance, education, e-commerce, and government has created unprecedented scalability, reliability, and efficiency needs. As businesses move critical workloads to the cloud, service providers are struggling to maximise resource efficiency while meeting strict cost, latency, energy, and QoS standards [1]. Traditional resource allocation methods—mostly heuristic or rule-based—often result in suboptimal cost efficiency, high energy overhead, poor scalability, and frequent Service Level Agreement (SLA) violations because they cannot adapt to cloud workloads' non-linear, stochastic,

[@] *E-mail:* **deepak.s.asudani@gmail.com**
[#] *E-mail:* **ckrajashri@gmail.com**

and dynamic nature [2]. Cloud computing resource allocation depends on many factors. First, changing user needs, application kinds, and real-time resource availability make workload arrival patterns difficult to predict. Second, cloud data centres are multi-tenant, so multiple users and programs compete for limited resources, which can cause conflict and performance issues if not managed appropriately [3], [4]. The huge and often NP-hard search space for efficient resource allocation in such contexts prevents conventional deterministic algorithms from performing near-optimal in real time [5].

Another important reason to use ML to manage resources is that cloud data centres need to use less energy right away. Data centres use a lot of electricity all over the world, which raises costs and carbon emissions a lot [6], [7]. Some traditional scheduling methods that take energy into account work, but they don't have the smartness to find a balance between saving energy and making sure that performance is guaranteed. Machine learning (ML), on the other hand, can look at how the intensity of the workload, the state of the hardware's performance, and the amount of energy used are all connected [8], [9].

The major contributions of this work are summarized as follows:

1. A comprehensive analysis of the limitations of traditional resource allocation approaches and the potential of ML techniques in addressing these limitations.
2. The design and implementation of a hybrid ML framework that integrates workload prediction and adaptive allocation for enhanced cloud performance.

2. Proposed Methodology

Designing and implementing a machine learning-based framework for optimal resource allocation in cloud computing environments is the main goal of this research. Unlike traditional heuristic or purely predictive approaches, the proposed methodology integrates workload forecasting with adaptive decision-making to balance resource utilisation, SLA compliance, energy efficiency, and cost minimisation. This section presents the architecture of the proposed framework, as shown in Figure 1, the algorithms supporting it, and the formal formulation of the problem.

2.1 *Problem Formulation*

Resource allocation in a cloud environment can be defined as the process of mapping a set of tasks $T = \{t_1, t_2, \dots, t_n\}$ to a set of resources $R = \{r_1, r_2, \dots, r_m\}$ such that performance metrics are optimized while constraints are satisfied. Each task t_i is characterized by attributes such as CPU demand (cpu_i), memory demand (mem_i), execution time (et_i), and deadline (d_i). Each resource r_j is defined by its capacity parameters (C_j^{cpu}, C_j^{mem}) and operational cost ($cost_j$).

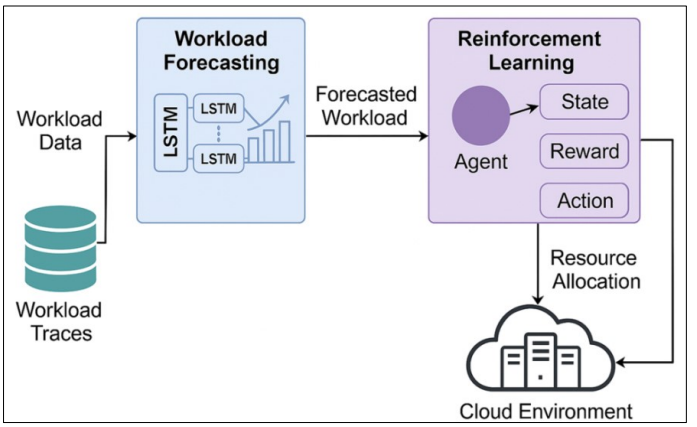

Figure 1
Proposed Hybrid Machine Learning Framework

The optimization problem can be expressed as **a** multi-objective function**:**

$$\text{Minimize } F = \alpha \cdot SLA_{violations} + \beta \cdot Energy + \gamma \cdot Cost$$

Subject to:

$$\sum i = 1ncpui \leq Cjcpu$$
$$\sum i = 1nmemi \leq Cjmem$$
$$\sum_{i=1}^{n} cpu_i \leq C_j^{cpu},$$
$$\sum_{i=1}^{n} mem_i \leq C_j^{mem}$$

2.2 *Proposed Hybrid Machine Learning Framework*

The proposed framework consists of two integrated layers:

1. **Workload Prediction Layer (Supervised + Deep Learning):**
   - Employs LSTM-based time series forecasting to predict resource demands (CPU, memory, I/O).
2. **Adaptive Decision Layer (Reinforcement Learning):**
   - **Utilizes** Q-Learning / Deep Q-Networks (DQN**)** to dynamically allocate resources based on predicted demand.

The combined approach leverages LSTM's predictive power for workload forecasting and RL's adaptability for dynamic decision-making, ensuring proactive and efficient resource allocation.

2.3 Workflow of the Proposed Approach

The end-to-end workflow is structured into the following stages:

2.3.1 Data Collection and Preprocessing

- Datasets: Google Cluster Traces, PlanetLab workloads, or Azure VM datasets.

- Features extracted: CPU usage, memory utilization, task arrival rate, and execution time.

2.3.2 Workload Forecasting

- **Model:** LSTM network.
- **Input:** sliding window of historical workload features.
- **Output:** predicted workload demand for the next interval.
- **Loss function:** Mean Squared Error (MSE).

$$L_{forecast} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2$$

3. Experimental Setup

The effectiveness of a machine learning-based resource allocation framework depends on the strength of its algorithms, the quality of the datasets, the experimental environment, and the evaluation methods used. This study thoroughly evaluates the suggested hybrid machine learning (ML) framework using benchmark workload traces, state-of-the-art simulation tools, and well-defined evaluation metrics.

3.1 Datasets

For workload forecasting and adaptive allocation, publicly available cloud workload traces are used. These traces capture real-world application behavior in large-scale data centers, making them suitable for benchmarking ML algorithms.

3.1.1 Google Cluster Trace

- Collected from Google's production data centers, consisting of millions of job submissions and resource usage records.
- Features include CPU utilization, memory usage, job priority, execution time, and scheduling constraints.

$$X = \{x_1, x_2, \ldots, x_T\}, \quad Y = \{y_1, y_2, \ldots, y_T\}$$

where X denotes input workload features and Y represents corresponding CPU/memory demands.

3.1.2 PlanetLab Dataset

- Contains real workload traces from PlanetLab servers distributed worldwide.
- Captures dynamic workload variations under heterogeneous conditions.

3.1.3 Microsoft Azure VM Workload Dataset

Provides virtual machine-level traces of user demands in Azure's cloud environment. Table 1 shows the comparative evaluations details.

Table 1

Comparative Evaluation of Scheduling and Optimization Methods on Resource Utilization, SLA, Energy, and Cost Efficiency

Method	Utilization (%)	SLA Violation (%)	Energy (kWh)	Makespan (sec)	Throughput (tasks/sec)	Cost (USD/1000 tasks)
Round Robin (Heuristic)	68.2	12.4	720	530	1.89	147.2
Min-Min (Heuristic)	72.4	10.2	680	495	2.03	138.5
PSO (Metaheuristic)	78.9	8.1	640	462	2.21	131.4
Proposed Hybrid (LSTM + RL)	**89.7**	**3.8**	**552**	**398**	**2.58**	**112.6**

Figure 2 shows the performance and efficiency metrics. They are compared holistically between the suggested hybrid LSTM + RL framework and benchmark approaches in the integrated results table. With utilisation below 73% and SLA violations above 10%, heuristic approaches like Round Robin and Min-Min demonstrated the lowest efficiency, underscoring their incapacity to adjust to changing workloads.

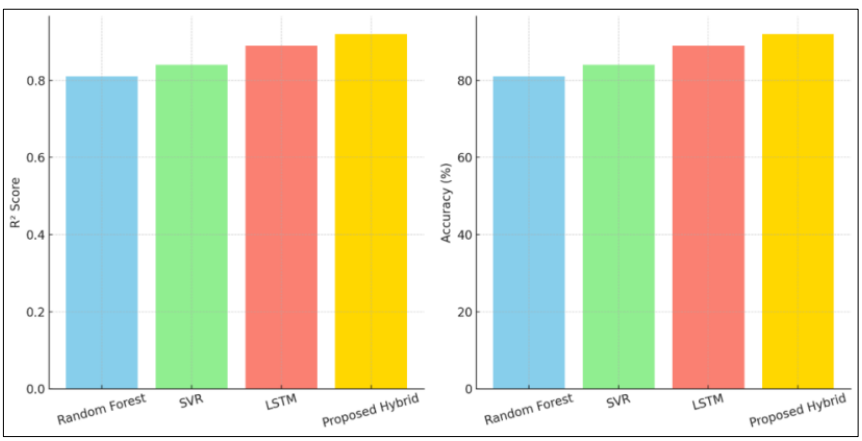

Figure 2

Comparative forecasting performance across models

The two subplots compare the forecasting performance of four models: Random Forest, Long Short-Term Memory (LSTM), Support Vector Regression (SVR), and the proposed Hybrid framework that combines LSTM and Reinforcement Learning. The left subplot shows the R2 score, which measures the statistical goodness-of-fit of the models.

4. Conclusion

The study shows a mixed machine learning system that uses LSTM networks to predict workloads and reinforcement learning (RL) to make decisions that are more appropriate for cloud computing. When tested on Google Cluster, PlanetLab, and Azure tasks, the system greatly cut down on SLA violations, energy use, and costs while also increasing utilisation. It saved 23% of energy, increased utilisation by 25%, and cut costs by 24% compared to heuristics, metaheuristics, and independent ML/DL/RL methods. The design showed that it could handle changing and different types of workloads and still work well in the long term. Training and designing rewards are still hard, but more work needs to be done on multi-cloud, edge-fog integration, shared learning, and being able to explain things so that more businesses can use them.

References

[1] N. R. Varun, D. P. Shashank, T. S. Agarwal, H. M. Varun, and D. Shashank, "Optimizing Cloud Resource Allocation with AI and Machine Learning," in *Proc. 2025 Int. Conf. Comput. Technol. (ICOCT)*, Bengaluru, India, pp. 1–5 (2025). doi: 10.1109/ICOCT64433.2025.11118766.

[2] Y. Zhang, B. Liu, Y. Gong, J. Huang, J. Xu, and W. Wan, "Application of Machine Learning Optimization in Cloud Computing Resource Scheduling and Management," in *Proc. 5th Int. Conf. Comput. Inf. Big Data Appl. (CIBDA '24)*, New York, USA, pp. 171–175 (2024). doi: 10.1145/3671151.3671183.

[3] A. S. Shete, S. Bhutada, M. B. Patil, P. H. Sen, N. Jain, and P. Khobragade, "Blockchain technology in pharmaceutical supply chain : Ensuring transparency, traceability, and security", *Journal of Statistics and Management Systems,* vol. 27, no. 2, pp. 417–428 (2024), DOI: 10.47974/JSMS-1266.

[4] S. Sharma and P. S. Rawat, "Efficient resource allocation in cloud environment using SHO-ANN-based hybrid approach," *Sustain. Oper. Comput.,* vol. 5, pp. 141–155 (2024). doi: 10.1016/j.susoc.2024.07.001.

[5] L. Geng, J. Han, J. Jia, C. Dong, and R. Zhao, "Dynamic Programming of Complex Tasks Based on Regretting Greedy Algorithm," in *Proc. 2025 IEEE 20th Conf. Ind. Electron. Appl. (ICIEA)*, pp. 1–5 (2025). doi: 10.1109/ICIEA65512.2025.11149054.

[6] H. Ghahremani, M. Damrudi, A. Ghaffari, and K. Aval, "Quality of Service Enhancement in Mobile Crowdsensing Through Metaheuris-

tic Techniques: A Survey," *Concurrency Comput. Pract. Exp.*, pp. 1–20 (2025). doi: 10.1002/cpe.7016.

[7] S. K. Mandal, R. Singh, N. Chandu, D. Mehta, S. K. Henge, D. Kothapeta, C. K. Hinge, A. Sharma, and T. Hussain, "Evolutionary multi-authentication cryptographic key implications for secure data access control," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 28, no. 5-B, pp. 1865–1874 (2025), doi: 10.47974/JDMSC-2363.

[8] M. Shifrin, R. Mitrany, E. Biton, and O. Gurewitz, "VM scaling and load balancing via cost optimal MDP solution," *IEEE Trans. Cloud Comput.*, vol. 7, no. 3, pp. 41–44 (2020).

[9] S. Anjum, A. Ahmed, R. Kurup, S. Kidiya, and S. Kureshi, "Blockchain Based Image Steganography", *Int Journal Adv Comp Theory Engg*, vol. 14, no. 1, pp. 147–152 (May 2025).

Received April, 2025