International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(23s) 2026 pp. 451-459
https://doi.org/10.70917/ijcisim-2026-5586
Article

ML-WDLB: A Machine Learning–Based Dynamic Load Balancing Algorithm for Efficient Resource Allocation in Cloud Computing

Jasobanta Laha1*, Sukanta Kumar Das[2], Soumya Samarpita[3], Lalmohan Pattnaik[4], Ritunsa Mishra[5]

[1,2,3,4,5]Faculty of Engineering & Technology (FET), Sri Sri University, Kataka, Odisha
Corresponding author : Jasobanta Laha (Email id : yeshmcp@gmail.com)

Abstract: Cloud computing allows flexible and scalable computing resources. However, it is still a challenging problem how to efficiently distribute incoming workloads onto Virtual Machines (VMs). If the tasks are not well distributed, some VMs can be heavily loaded and others can be idle or lightly loaded. This can increase response time and processing time and can also reduce the overall utilization and performance of the cloud system. This paper presents a Machine Learning–Based Weighted Dynamic Load Balancing (ML-WDLB) approach for making VM selection more responsive to changing workload conditions. The approach uses both historical and current VM information, including CPU utilization, memory utilization, processing time, response time, and bandwidth. A Random Forest model is used to estimate the suitability of a VM for an incoming cloudlet or task. This prediction is combined with a weighted VM score that reflects the current condition of each VM. The VM with the highest final score, provided that it is below the specified load threshold, is selected for task execution. The performance of the approach can be assessed using response time, processing time, makespan, throughput, resource utilization, and load imbalance. The main aim of ML-WDLB is to provide a practical balance between prediction and current system information so that workload can be distributed more effectively as cloud conditions change.
Keywords: NA

1. Introduction

Cloud computing has become a widely used computing model for delivering computing resources on demand through distributed data centers and Virtual Machines (VMs) [1]. As more applications move to cloud platforms, the workload received by these systems can change considerably over time. A lightly loaded VM at one point in time may be heavily loaded when a sudden group of tasks arrives. For this reason, effective workload distribution is important for maintaining resource utilization and Quality of Service(QoS)[2,3]. Load balancing is responsible for distributing incoming tasks among available VMs so that the workload does not become concentrated on a small number of resources. A good load-balancing mechanism should not only avoid overloaded VMs but should also make reasonable use of VMs that have available capacity. In practice, many of the traditional approaches rely on predefined rules, fixed thresholds or the current state of a VM. Such decisions may have less effect when the conditions of workload change frequently [4].

Alternatively, ML can be used to take into account previous performance information when making VM-selection decisions. Therefore, recent work has investigated ML, deep learning and reinforcement learning techniques for resource management and workload scheduling [5]. Parameters such as CPU utilization, memory utilization, processing time, response time and bandwidth can provide useful information about the state of a VM and its capacity to handle another task.

According to this idea, this research proposes the Machine Learning–Based Weighted Dynamic Load Balancing (ML-WDLB) approach. The method estimates the VM suitability with a Random Forest model and combines the prediction with a weighted score calculated from the current VM condition. The idea is to base a VM selection on a combination of learned knowledge and instantaneous measurements rather than a single parameter. The proposed approach is to reduce response time, processing time, makespan and load imbalance and improve throughput and utilization of resources.

Traditional load-balancing techniques are usually based on predefined rules, thresholds, or current VM states that may not react effectively in the case of continuously changing workloads [4]. Therefore, recent research has looked into Machine Learning (ML) and intelligent prediction techniques to identify workload patterns and improve resource allocation decisions [5]. ML-based methods can evaluate parameters including CPU utilization, memory utilization, processing time, response time, and bandwidth to predict VM suitability prior to task allocation.

In this paper, we propose a Machine Learning–Based Dynamic Load Balancing (ML-DLB) approach which employs ML based performance prediction of VMs and weighted VM scoring to select the best VM for scheduling arriving cloudlets. The proposed approach is aimed at reducing the response time, processing time, makespan, and load imbalance; while improving throughput and resource utilization. The approach promises to be more effective for achieving good cloud performance with regard to the dynamic nature of workload in clouds.

2. Literature Survey

1. Sharmah and Bora [6] analyzed the dynamic load-balancing methods applied in cloud computing and discussed the challenges related to the allocation process of the load to virtual machines. The article under consideration classifies load-balancing approaches in terms of dynamic and static ones and describes different criteria, including response time, resource utilization, and workload distribution. In addition, it explains why dynamic load-balancing is more effective in clouds where the state of VMs may change constantly.

2. While discussing dynamic load balancing in cloud computing, Laha et al. [7] propose a new approach to the problem. The authors address the importance of response and execution time, as well as the stability of the system. The results of the research suggest that the choice of a proper VM (virtual machine) and the balance of the workload can affect the performance of the entire system significantly. In addition, the article highlights the efficiency of utilizing multiple VM performance metrics in the process of load balancing.

3. Zhou et al. [8] conducted a study on resource-scheduling strategies based on Deep Reinforcement Learning (DRL) in a cloud computing environment. The work illustrates the application of DRL in learning scheduling policies that are able to adapt to the constant changes in the cloud computing environment, which makes them suitable for solving resource allocation problems in such a scenario. However, the use of DRL may involve significant computational costs during training, implementation, and optimization, which makes ML approaches seem more appealing and efficient in some cases, particularly when the task at hand only requires a well-chosen machine learning model for selecting the best VMs for a given set of tasks.

4. Khan [9] presented a dynamic load-balancing framework which is based on deep learning, reinforcement learning, and hybrid optimization. The CNN-RNN model was applied for VM-load prediction, while scheduling was supported by reinforcement learning and optimization. After conducting experiments with Cloudsim, the results demonstrated that the proposed approach can improve the resource utilization, makespan, and CPU utilization. The paper proves the effectiveness of applying the learning-based methods, yet the combination of multiple approaches makes the solution more complex than some existing ones.

5. Zhanuzak et al. [10] presented an Enhanced Dynamic Load Balancing (EDLB) algorithm for cloudlet assignment. This paper is significant because EDLB shifts the VM-placement decision-making to the time of task allocation instead of being based on static selection or relying on migration at a later stage. The study demonstrated how dynamic VM placement can improve the execution performance while minimizing the negative impact of over-loaded and under-loaded VMs.

6. Sanjalawe et al. [11] proposed a smart load-balancing framework that uses the combination of feature selection and advanced deep-learning models to create an innovative approach to address the problem. The paper states that based on the Google Cluster Trace dataset, the proposed solution demonstrated better throughput, makespan, resource utilisation, and energy consumption than the existing solutions. The authors concluded that data-driven methods can be used to make better decisions in a dynamic cloud environment.

3. Pseudocode: ML-WDLB (Machine Learning–Based Weighted Dynamic Load Balancing)

Input:

C = Set of incoming Cloudlets/Tasks

V = Set of available Virtual Machines (VMs)

D = Historical VM performance dataset

T = Maximum allowable VM load threshold

w1, w2, w3, w4, w5 = Weights for VM parameters

α = Weight of ML prediction

Output:

VM assignment for each Cloudlet

Begin

1. // Collect historical VM performance data

D ← Collect historical CPU, Memory, Bandwidth,

Processing Time, Response Time and VM Load data

2. // Preprocess the historical dataset

D ← Remove missing or invalid values

D ← Normalize the VM performance parameters

3. // Train the Machine Learning model

ML_Model ← Train Random Forest Model using D

4. // Initialize VM status

```
For each VM_i ∈ V do
```

Obtain current CPU utilization

Obtain current Memory utilization

Obtain current Bandwidth

Obtain current Processing Time

Obtain current Response Time

Obtain current VM Load

End For

5. // Process each incoming Cloudlet

```
For each Cloudlet C_j ∈ C do
  BestVM ← NULL
  BestScore ← -∞
```

6. **// Evaluate all available VMs**

```
    For each VM_i ∈ V do
```

7. **// Check whether the VM is overloaded**

```
      If VM_Load_i ≥ T then
      Continue to next VM
    End If
```

8. **// Obtain current VM parameters**

```
      CPU_i ← Current CPU Utilization of VM_i
      MEM_i ← Current Memory Utilization of VM_i
      BW_i  ← Current Bandwidth of VM_i
      PT_i  ← Current Processing Time of VM_i
      RT_i  ← Current Response Time of VM_i
```

9. **// Normalize current VM parameters**

```
      CPU_i ← Normalize(CPU_i)
      MEM_i ← Normalize(MEM_i)
      BW_i  ← Normalize(BW_i)
      PT_i  ← Normalize(PT_i)
      RT_i  ← Normalize(RT_i)
```

10. **// Predict VM suitability using the trained ML model**

```
      MLPred_i ← ML_Model.Predict(
              CPU_i, MEM_i, BW_i, PT_i, RT_i)
```

11. **// Calculate weighted VM suitability score**

```
      VMScore_i ←
        w1 × (1 - CPU_i) +
        w2 × (1 - MEM_i) +
        w3 × (1 - RT_i) +
        w4 × (1 - PT_i) +
        w5 × BW_i
```

12. **// Combine ML prediction and weighted VM score**

FinalScore_i ←

α × MLPred_i +

(1 - α) × VMScore_i

13. // Select the VM with the highest suitability score

If FinalScore_i > BestScore then

BestScore ← FinalScore_i

BestVM ← VM_i

End If

End For

14. // Check whether a suitable VM is available

If BestVM ≠ NULL then

15. // Assign the cloudlet to the selected VM

Assign Cloudlet C_j to BestVM

16. // Update the selected VM's load

Update CPU utilization of BestVM

Update Memory utilization of BestVM

Update Bandwidth of BestVM

Update Processing Time of BestVM

Update Response Time of BestVM

Update VM Load of BestVM

Else

17. // No suitable VM is currently available

Place Cloudlet C_j in Waiting Queue

End If

18. End For

19. // Update the ML training dataset with new observations

Record actual VM performance after task execution

20. // Retrain the ML model periodically when sufficient

// new performance data becomes available

If Retraining Condition = TRUE then

Update dataset D

Retrain ML_Model

End If

21. Return VM assignments

End

4. Mathematical Formulation

For each available VM i, the **weighted suitability score** is:

$$\mathbf{VMScorei=w1(1-CPUi)+w2(1-MEMi)+w3(1-RTi)+w4(1-PTi)+w5(BWi)}$$

where:

$$\mathbf{w1+w2+w3+w4+w5=1}$$

The final decision score is:

$$\mathbf{FinalScorei=\alpha\times MLPredi+(1-\alpha)\times VMScorei}$$

where:

- MLPredi = suitability predicted by the Random Forest model.
- VMScorei = current weighted VM suitability.
- α = importance given to ML prediction.
- $1-\alpha$ = importance given to current VM condition.

The selected VM is:

$$\mathbf{BestVM=\arg\max_{VMi\in V} FinalScorei}$$

subject to:

$$\mathbf{VMLoadi<T}$$

5. Literature-Based Comparison

Sl. No.	Algorithm / Study	Year	ML / AI Technique	Response Time	Processing / Execution Time	Makespan	Overall Observation
1	Sharmah & Bora – Dynamic LB Survey	2023	Traditional Dynamic LB	Medium	Medium	Medium	Survey of dynamic techniques; identifies response time and resource utilization as important measures
2	Laha et al. – Dynamic LB	2024	Dynamic / Rule-based approach	**Low–Medium**	**Low–Medium**	Medium	Focuses on response time, execution time and system stability

3	Zhou et al. – DRL-based Scheduling	2024	Deep Reinforcement Learning	**Low**	**Low**	**Low**	DRL provides adaptive scheduling but has higher model complexity
4	Khan – RL + CNN-RNN + HLFO	2024	DL + RL + Hybrid Optimization	**Low**	**Low**	**Very Low**	Uses CNN-RNN load prediction, RL clustering and multi-objective optimization
5	Zhanuzak et al. – EDLB	2024	Enhanced Dynamic LB	**Low**	**Low**	**Low**	Dynamically selects cloudlet placement and focuses strongly on makespan and execution time
6	Sanjalawe et al. – SLADRO	2025	Feature Selection + Deep Learning	**Low**	**Low**	**Low**	Reported improvements in throughput, makespan, resource utilization and energy efficiency
Proposed	**ML-WDLB**	**Proposed**	**Random Forest + Weighted VM Scoring**	**Expected Very Low**	**Expected Low**	**Expected Low**	Predictive VM selection using CPU, memory, bandwidth, response time and processing time

6. Discussion of Literature-Based Comparison

The reviewed articles demonstrate that the paradigm of cloud load balancing is shifting from the traditional dynamic rule-based approaches to the data-centric ones, which employ machine learning, deep learning, and reinforcement learning methods. Notably, traditional techniques are typically easier to implement, while the ones relying on learning algorithms offer the ability to leverage historical and real-time data to make more accurate predictions. The reviewed works [1]-[6] report that intelligent allocation of workloads can lead to improved response time, reduced time of servicing and shorter makespan. At the same time, the complexity of some of the considered above approaches suggests that there is a need to identify a less resource-consuming ML approach for VM selection.

Sharmah and Bora [1] primarily focus on conventional dynamic load-balancing approaches, which are based on the instantaneous loading and the available resources of a virtual machine (VM). The former is relatively simple and efficient, provided that the system under study is stable. The critical limitation of such a method is that it cannot properly account for the impact of additional tasks on the performance of a VM. Laha et al. [2] highlight the response time, execution period, and load-balancing parameters, which are used in the dynamic VM selection. However, the performance of the system may significantly decrease if the changes in the instantaneous loading are frequent and dynamic.

Zhou et al. [3] conducted a study on the use of Deep Reinforcement Learning (DRL) for cloud resource scheduling. The main benefit of the proposed approach is that an agent can learn optimal scheduling policies through interactions with the environment. Such an approach allows a higher degree of adaptation as compared to the standard methods. However, it also requires extensive training data and interactions with the environment and may involve high computational costs, which may add to the overhead of the load-balancing system.

Khan [4] used a combination of CNN-RNN, Reinforcement Learning, and hybrid optimization. Such a combination can provide strong prediction and optimization capabilities and has the potential to improve resource utilization and makespan. The trade-off is a more complicated architecture. Training, tuning, and maintaining several learning and optimization components can also add computational cost to the overall system.

Zhanuzak et al. [5] proposed an Enhanced Dynamic Load Balancing (EDLB) method that dynamically determines an appropriate VM for task placement. The approach focuses on execution time and makespan and improves the allocation process by making the placement decision during task scheduling. However, its VM-selection process does not use a machine-learning model to learn from historical VM behavior.

Sanjalawe et al. [6] used feature selection together with advanced deep-learning models for cloud load balancing. Their results show the potential of data-driven techniques to improve throughput, makespan, resource utilization, and energy efficiency. Nevertheless, deep-learning models can require considerable training data and computing resources. This may make them less convenient when a relatively lightweight solution is preferred.

7. Why ML-WDLB is Better?

The proposed ML-WDLB is designed to keep the useful predictive capability of machine learning without introducing the level of complexity found in some hybrid deep-learning and reinforcement-learning approaches. The method collects historical and current VM information and uses a Random Forest model to estimate the suitability of each VM. The prediction is then combined with a weighted score based on CPU utilization, memory utilization, response time, processing time, and bandwidth.
This combination is important because the VM decision is not made from the current workload alone. Historical information contributes to the prediction, while the weighted score reflects the VM's present condition. In this way, the method considers two aspects of the scheduling problem at the same time: what the model has learned from earlier observations and what is happening in the VM at the moment of task allocation.

The main difference is therefore the basis on which the VM is selected. Instead of considering only the present workload, ML-WDLB combines predicted VM suitability with the current condition of the VM. This allows the selection process to respond to changes in resource usage while still making use of information learned from earlier observations.

The final decision is calculated as:

$$FinalScore_i = \alpha \times MLPred_i + (1-\alpha) \times VMScore_i$$

where the weighted VM score is:

$$VMScore_i = w_1(1-CPU_i) + w_2(1-MEM_i) + w_3(1-RT_i) + w_4(1-PT_i) + w_5(BW_i)$$

The VM with the highest **FinalScore** and satisfying the load threshold is selected for task execution.

8. Conclusion

The ML-WDLB algorithm presented in this work provides a practical approach to dynamic workload distribution in cloud computing. Traditional dynamic load-balancing methods [1,2] generally make decisions from the current VM state, while more advanced DRL, hybrid DL/RL, and deep-learning approaches [3–6] can provide stronger learning capabilities at the cost of greater model and implementation complexity. ML-WDLB takes a middle path by combining Random Forest–based VM suitability prediction with a weighted multi-parameter scoring method. The suggested approach takes into account the CPU utilization, memory utilization, bandwidth, response time, and processing time when choosing the best VM. It discards the VMs that have already hit the specified level of load and aggregates the ML forecast data and the weighted scores of the rest to determine the most suitable one. The result has better chances to be successful compared to an approach where the current state of these parameters is used outright. The approach is expected to reduce the response time, processing time, makespan, and load imbalance while improving throughput and resource utilization. More importantly, it offers a certain degree of transparency since it is possible to identify VMs' scores based on the final ML prediction and respective VM parameters. However, the practical efficiency of ML-WDLB requires validation in CloudSim or CloudAnalyst using the same VMs setup, workload description, and performance evaluation as in the case of other algorithms under investigation. This way, it will be possible to obtain unbiased results for comparison and consider whether the proposed approach is indeed beneficial for cloud resource provisioning.

REFERENCES

1. P. Mell and T. Grance, *The NIST Definition of Cloud Computing*, NIST Special Publication 800-145, National Institute of Standards and Technology, 2011.
2. R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility," *Future Generation Computer Systems*, vol. 25, no. 6, pp. 599–616, 2009.
3. N. R. Herbst, S. Kounev, and R. Reussner, "Elasticity in Cloud Computing: What It Is, and What It Is Not," *Proceedings of the 10th International Conference on Autonomic Computing*, pp. 23–27, 2013.
4. M. Randles, D. Lamb, and A. Taleb-Bendiab, "A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing," *Proceedings of the IEEE 24th International Conference on Advanced Information Networking and Applications Workshops*, pp. 551–556, 2010.
5. J. M. Shah, A. A. Shah, and others, research on intelligent and ML-based resource management and load balancing in cloud environments, 2021.
6. D. Sharmah and K. C. Bora, **"A Survey on Dynamic Load Balancing Techniques in Cloud Computing,"**E2A2022, 2023.
7. J. Laha, S. Pattnaik, and K. S. Chaudhury, **"Dynamic Load Balancing in Cloud Computing: A Review and a Novel Approach,"***EAI Endorsed Transactions on Internet of Things*, vol. 10, no. 1, 2024.
8. G. Zhou, W. Tian, R. Buyya, R. Xue, and L. Song, **"Deep Reinforcement Learning-Based Methods for Resource Scheduling in Cloud Computing: A Review and Future Directions,"***Artificial Intelligence Review*, vol. 57, 2024.
9. A. R. Khan, **"Dynamic Load Balancing in Cloud Computing: Optimized RL-Based Clustering with Multi-Objective Optimized Task Scheduling,"***Processes*, vol. 12, no. 3, 519, 2024.
10. R. Zhanuzak, M. A. Ala'anzy, M. Othman, and A. Algarni, **"Optimising Cloud Computing Performance with an Enhanced Dynamic Load Balancing Algorithm for Superior Task Allocation,"***IEEE Access*, 2024.
11. Y. Sanjalawe, S. Fraihat, S. Al-E'mari, M. Abualhaj, S. Makhadmeh, and E. Alzubi, **"Smart Load Balancing in Cloud Computing: Integrating Feature Selection with Advanced Deep Learning Models,"***PLOS ONE*, vol. 20, no. 9, e0329765, 2025.