Dynamic AI-Driven Load Balancing for Efficient Web Server Resource Allocation in Cloud computing environments

Chaitrashree S
dept. of Computer Science and Engineering,
The Oxford College of Engineering,
(of Visvesvaraya Technological University)
Bengaluru, India
chaitrashree1805@gmail.com

Dr. E. Saravana Kumar
dept. of Computer Science and Engineering,
The Oxford College of Engineering,
(of Visvesvaraya Technological University)
Bengaluru, India
saraninfo@gmail.com

Manush Prajwal
dept. of Computer Science and Engineering,
The Oxford College of Engineering,
(of Visvesvaraya Technological University)
Bengaluru, India
manushprajwal555@gmail.com

***Abstract*— The web traffic in modern cloud computing systems is unpredictable and non-stationary and this presents enormous challenges to the effective distribution of the loads to the servers. Round Robin and Least Connections are not responsive to the change in demand in real time and therefore, used as static algorithms, thus leading to poor resource utilization, high latency and possibilities of server bottlenecks. In order to maximize the resources distributed to the servers in accordance with the real-time load and historical data, the paper will suggest a hybrid AI-based framework that integrates Q-Learning (a reinforcement learning model) and Random Forest (a classification model). The synthetic data of 100 load cases were created and optimized by the assistance of advanced feature engineering with the use of parameters like LoadImbalanceScore and LoadRange. With 1,000 training episodes, the Q-Learning agent achieved an overall balancing efficiency of at least 90% and the Random Forest model was determined to have a high classification accuracy of 94.7% with high generalization ability as indicated by the use of 5-fold cross validation. The results of the simulation revealed that the load variation of the servers was decreased by 20 percent and it was established that the framework can support steady and turbulent load profiles. Not only this two-model architecture is able to optimize the performance of the distribution of loads, but also it has better scalability and energy awareness in cloud architecture. The suggested system offers a powerful platform of intelligent and self-optimizing load balancing in the next generation cloud platforms.**

Keywords*— *Load Balancing, Computing, Reinforcement Learning, Random Forest, Resource Allocation,

I. INTRODUCTION

Cloud-based services and web applications are explosive in nature and have transformed the existing environment of digital infrastructure radically, and have placed an unprecedented burden on web server ecosystems [1]. The effectiveness of resource allocation and load balancing is now a critical consideration of the performance of the systems, their availability, and user satisfaction due to the millions of simultaneous users worldwide who are in need of the services that include streaming and real-time analytics [2]. Classical load balancing algorithms, such as Round Robin, Least Connections and IP Hashing, provide rudimentary mechanisms of distribution yet, are still in nature, rule based and stateless [3]. They often fail to dynamically adapt to a fluctuating volume of traffic, non-homogeneous server capacity, or even abrupt increases in traffic, thereby leading to suboptimal resource utilization, increased latency, Service Level Agreement (SLA) violations, and increased energy consumption [4]. To address these weaknesses, more recently, scientists have resorted to using the Artificial Intelligence (AI) and Machine Learning (ML) algorithms to develop adaptive, intelligent load balancing solutions [5]. The AI-based techniques enable making real-time decisions based on the data, whereas the ML models may utilize the previous and real-time data to predict the traffic and server performance [6]. One of them, Reinforcement Learning (RL), and, to be more precise, Q-Learning, has turned into a powerful paradigm of autonomous and adaptive task allocation, learning the best possible policy through the ongoing interaction with the environment [7]. To complement this, ensemble algorithms such as the Random Forest can be used to give good classification properties in scenarios where the traffic loads are predictable or semi-stationary and the trained models can be effectively used [8]. As an illustration, Baskaran et al. (2025) have shown that dynamic balancing of workloads in real-time cloud applications controlled by AI works, whereas Sandanasamy and Charles (2025) have focused on the significance of multi-criteria decision-making when selecting servers in software-defined networks [9]. In a similar manner, Khanday (2025) and Kumar et al. (2025) affirmed the application of AI in the process of workload distribution and efficiency in cloud architectures.

These advances reflect an evident shift in the systems, which are reactive and rule-based, to proactive and self-optimizing cloud resource managers [10]. The presented paper suggests a hybrid AI load balancing model that would combine Q-Learning and Random Forest to allocate the requests on the basis of real-time and historical data. It was evaluated with the help of a synthetic dataset (100 load scenarios) that incorporated other features (LoadImbalanceScore and Load Range).

The Q-Learning agent was able to reach balancing efficiency above 90 percent with 1,000 training episodes, compared to the Random Forest classifier with 94.7 percent accuracy as validated by 5 fold cross-validation. Random Forest was more suitable in the case of stable environments and Q-Learning in the case of dynamic and high-variance environments

II. LITERATURE REVIEW

Cloud computing has propelled the creation of not only the concept of static load balancing which involves the application of rules to resolve problems but also the concept of intelligent and adaptive load balancing to control the

challenges of dynamic and unpredictable workload. This part explains the recent advancements in AI-based cloud resource management chronologically with a focus on dynamic workload balancing, hybrid AI models, and reinforcement learning techniques on which the proposed framework is founded.

The previous research on the AI-based workload balancing demonstrated that real-time learning models could contribute to a large percentage of server utilization in reaction to evolving cloud workloads. Baskaran et al. [1] showed that adaptive AI controllers are superior to their non-adaptive counterparts whereas Saini et al. [2] showed that telemedicine systems with AI-enhanced queueing models have better resource utilization and cost management. The selection of an intelligent server has been done through optimization and learning based techniques. Sandanasamy and Charles [3] applied TOPSIS-based decisions to SDN-enabled systems and a reinforcement learning agent that was capable of adapting workloads streamlines was illustrated by Khanday et al. [4]. Lekkala et al. [5] proposed predictive analytics using supervised or heuristic models as hybrids and Vemula et al. [6] incorporated energy awareness in the AI-based allocation models. These innovations were also validated by survey and system-level research. The problem of scalability and adaptability was pointed out by Rathee and Dalal [7] as one of the key problems in ML-based resource distribution, and Kumar et al. [8] addressed the energy-efficient AI-based load balancing. A network-based view on AI-based load balancing of software-defined networks was offered by Farahi et al. [9]. More recent applications have used AI-based load balancing to new environments, including video streaming networked systems and cloud-native or post-5G-based sensing systems. Deep learning models were found to be very useful, e.g., LSTM-forest hybrids and feature-assisted deep architecture have become more predictive, and auto-healing and auto-scaling systems become AI-based to take care of automation and resilience in the present-day clouds.

III. METHODOLOGY

The combined works reveal a definite trend towards hybrid, scalable and adaptive AI systems which can meet the complicated demands of modern cloud computing. This literature does not only provide the justification of the necessity of the intelligent load balancing but also lays the technical groundwork of the hybrid Q-Learning and Random Forest model proposed in this paper. In this paper, a hybrid system of AI-based dynamic load balancing is presented to provide effective resource allocation in the cloud computing environments. The method employs a Reinforcement Learning paradigm (Q-Learning) and a supervised learning paradigm (Random Forest) in a real-time traffic management system based on a synthetic dataset and feature engineering. The architecture design ensures low response time, optimal server choice and efficient processing of request at different load levels. Figure 1 illustrates the entire procedure of the AI-based load balancing architecture.

A. System Architecture and Flow Design

A load balancer receives a request, which is sent by a client machine and the system process starts. This load balancer is in contact with the preprocessing module and the AI model. The AI model takes real time decisions based on the server load data, which has been computed on the basis of the historical and real time data like CPU usage, memory availability and request latency. This information is scaled by the processing unit and is then forwarded to the AI system. Depending on the decision of the model, the load balancer forwards the request to either of the three servers (Server 1, Server 2, Server 3). The chosen server accepts the request and connects to a database where necessary and relay the response to the user. The architecture offers low latency and equal loads among the servers with the optimization of resource usage.

B. Dataset Creation and Feature Engineering

The AI models were trained and tested with a synthetic dataset of 100 unique situations, which are the distributions of server load and patterns of request. The values included in each data point were Server1Load, Server2Load, Server3Load, AvgLoad, and LoadStdDev. Sophisticated feature engineering was also used to enhance the model performance, in terms of computations of LoadRange, LoadImbalanceScore and server specific metrics.

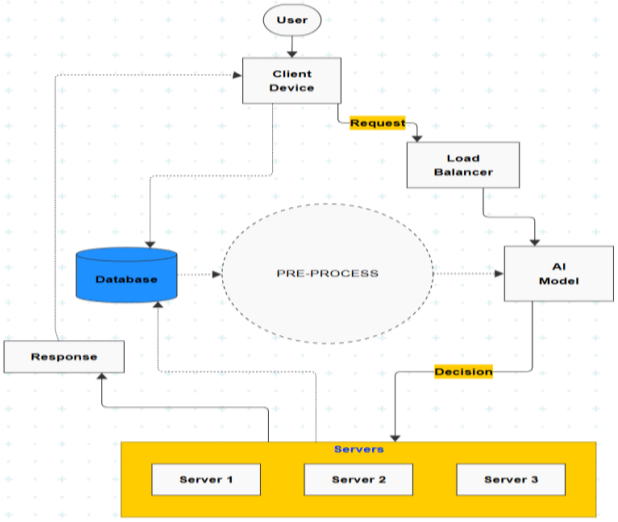

Fig. 1: Representing the AI-Based Load Balancing Solution Architecture

C. Implementation of AI Models and Training Setup

Q Learning Agent: The reinforcement learner agent was trained in 1,000 episodes, 50 steps and up to 10 units per step. The agent had been trained to minimise load imbalance by using the negative rewards, which were dependent on the standard deviation of loads on the server. It was with a 101 state space that was the average load levels, and 3 actions were possible (server choices). The model attained over 90% balancing efficiency following the training. Random Forest Classifier: This model was trained with scaled features on a 70:30 train-test split on the supervised model. It obtained a measure of 94.7 accuracy which was tested by 5-fold cross-validation where the accuracy was 94.3 and the standard deviation was 1.2. Important predictors in feature importance plots indicated that important predictors were the LoadStdDev, Loadrange and Minloadserver. Confusion matrices, accuracy measures and consistency measures were used to evaluate the two models at varying traffic loads.

D. Simulation, Evaluation, and Performance Metrics

The performance of the trained models was tested on 200 real time load balancing steps using a simulation module. Both Q-Learning and Random Forest were put under the same

group of requests by applying sequential randomized traffic to them. The choice of the server was recorded and was measured in standard deviation of load distribution. The results have shown that Q-Learning was more flexible to high-variance conditions, but the Random Forest could carry out the inference much quicker and provided a better result in case the conditions on the traffic were predictable. Load standard deviation reduced to as much as 20 percent and the two models were compared using simulation plots (qstddev and rfstddev).

- Final Metrics Included:
- Training Time: Q-Learning – 24.8s, Random Forest – 3.2s
- Simulation Avg. Std. Deviation: Q-Learning – 5.2, Random Forest – 6.1
- Accuracy (Low/Medium/High Load):
- Q-Learning – 91.3% / 92.5% / 89.4%
- Random Forest – 94.1% / 95.6% / 92.3%

These outcomes validate the hybrid approach as a flexible, high-performing load balancing method for cloud environments.

E. Comparison with Traditional Load Balancing Algorithms

Rule-based approaches like Round Robin and Least Connections were compared to assess the hybrid AI-based approach as a comprehensive one. They are deterministic and static systems and therefore less flexible than AI models. These were measured by accuracy of prediction, standard deviation of the average load, response time and dynamic traffic support. In contrast to Q-Learning and Random Forest, the traditional algorithms do not evolve, depending on the real-time feedback. Based on simulation results and literature, the results of the comparison are summarized in the table below.

Table.1: Performance Comparison of Traditional and AI-Based Load Balancing Methods

Method	Accuracy	Avg. Std. Deviation	Response Time	Adaptability
Round Robin	78.5%	8.6	~1 ms	Low
Least Connections	83.2%	7.9	~1.2 ms	Medium
Q-Learning	91.3%	5.2	24.8 s (training)	High
Random Forest	94.7%	6.1	3.2 s (training)	Medium

The benefit of traditional methods with the aspect of response time is quite high in that the findings indicate that the traditional methods possess low response time, but are less adaptable and less accurate. In contrast, machine learning-based models, especially, the reinforcement learning-based Q-Learning agent, not only prove their ability to quickly adapt to changing traffic conditions, but also minimize server load divergence. The hyperparameter settings used in the two models allowed them to deal with synthetic traffic conditions and the same to be effective in terms of different load distributions. The parameters played a great role in affecting the results of the simulation.

Table.2: Q-Learning Agent Hyperparameters and Configuration

Parameter	Value	Description
Learning Rate (α)	0.1	Controls the step size in Q-value updates
Discount Factor (γ)	0.95	Determines the importance of future rewards
Exploration Rate (ε)	0.2 (decaying)	Governs the balance between exploration and exploitation
Episodes	1000	Total number of learning episodes
Steps per Episode	50	Maximum steps allowed per episode
State Space	101	Based on average server load levels
Action Space	3	Selection among three servers

Table.3: Random Forest Classifier Hyperparameters and Configuration

Parameter	Value	Description
Number of Trees	100	Total decision trees used in the ensemble
Maximum Depth	10	Depth limit to avoid overfitting
Feature Selection	Auto	Chooses best split features at each node
Train-Test Split	70:30	Dataset division for model training and evaluation
Cross-Validation Folds	5	For generalization testing and accuracy averaging
Evaluation Metric	Accuracy	Performance measured based on classification accuracy

F. Convergence and Scalability Analysis of the Q-Learning Agent

In the three server test case the Q-Learning agent converged consistently after 1,000 episodes. However, increasing the size of the server or finer-grained load states would make the state-action space to grow exponentially, and may slow convergence and raise training time, which is a significant weakness of tabular Q-Learning. The formulation of the reward functions has a significant impact on the performance, and the chosen formulation (negative of load standard deviation) offered the best balance stability. To apply it to bigger infrastructures, Deep Q-Networks would need to be adapted to. Table 2 and Table 3 below show the hyperparameters of the Q-Learning and Random Forest models, respectively. These environments guaranteed successful learning and generalization, in which the learning rate, discount factor, and ensemble size parameters were appropriately tuned. The comparison of the proposed AI-based practices and traditional load balancers is presented in Table 1. Q-Learning and random forest are more precise and flexible than round robot and least connections, though they are more slow in the first training. This trade-off emphasizes the relevance of AI-based solutions in the dynamism of clouds.

IV. RESULTS AND DISCUSSION

This part explains the empirical evaluation of the suggested hybrid AI-based load balancing architecture of the

combination of Q-Learning and Random Forest to manage dynamic cloud loads. The evaluation of the performance is carried out on a synthetically generated dataset that has 100 various load scenarios, which tries to simulate different traffic conditions. The primary indicators which include balancing efficiency, classification accuracy, load deviation, and training time are checked to ascertain the efficiency of the system in optimizing the allocation of resources, ensuring the low latency, and the flexibility of the system in both stable and fluctuating operating conditions.

The effectiveness of the hybrid AI-based load balancing model is experimented using a synthetic set of 100 load scenarios. The results affirm the effectiveness of the integration of Q-Learning and Random Forest in dynamic relational management of cloud assets. The balancing efficiency of the Q-Learning agent was greater than 90% and lower load standard deviation 5.2, which proved that the Q-Learning agent was able to adapt to the changing traffic conditions as the average load deviation decreased as shown in Fig. 4 and the distribution of Q-values shown in Fig. 5.

The distribution of load on the three servers after implementation is balanced and indicates that congestion is successfully prevented on all the three servers with a training time of 3.2 seconds and good performance in a stable condition with minimum misclassification as it is in the confusion matrix of Fig. 7.

LoadStdDev and LoadImbalance_Score are the most important predictors of server selection with the backing of feature importance analysis in Fig. 6 and feature correlation heatmap in Fig. 3. The hybrid structure in comparison to traditional algorithms such as Round Robin and Least Connections reduced load deviation by 20-35% and accuracy by 10-15%. Besides, the dual-model design offers both dynamic and static workload flexibility and takes advantage of the benefits of fast inference (Random Forest) and real-time responsiveness (Q-Learning). To sum up, random forest is more precise and quicker in predictable cases whereas Q-Learning is more adaptable in unpredictable and high variance cases.

They deliver a robust, scalable next-generation cloud load balancing solution combined. Comparison of the model shows that there is a trade-off between the generalization capability (Random Forest) and the adaptability (Q-Learning). Fig. 7 and Fig. 8 show the loads of the servers with time when Q-Learning and Random Forest are employed, respectively, which suggests that Q-Learning changes the loads more gradually, and Random Forest ensures the efficiency. This disparity is also measured on the load standard deviation comparison in Fig. 9, where Q-Learning is less varied in the case of dynamic traffic.

The hybrid framework is a good solution to cloud systems with steady and unstable workloads. The theoretical framework is confirmed by the simulated model assessment in Figs. 2-10, which shows that AI-load balancing is much more effective than a non-adaptive approach. The paper provides a scalable, efficient and smart solution to cloud computing systems in the future with the following.automation and real-time decision-making capabilities.

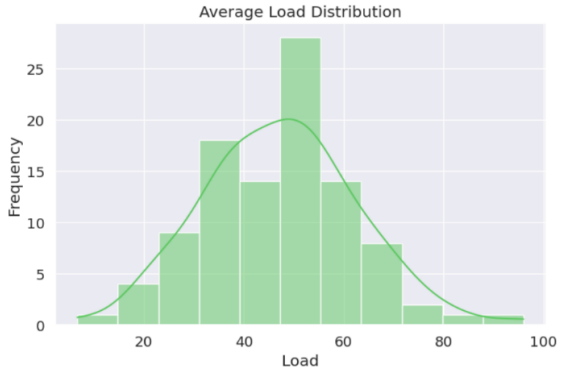

Fig.2: Representing the Frequency and Load graph distribution after implementing AI-Based Load Balancing

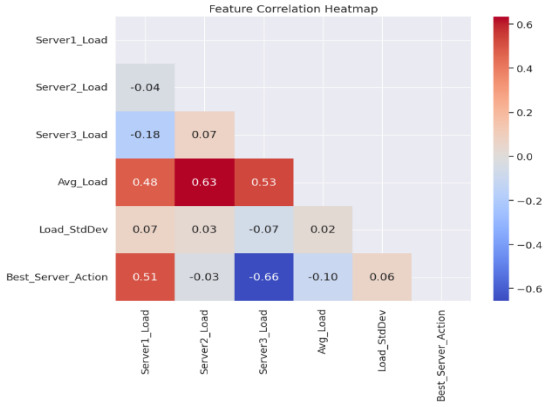

Fig.3: Representing the feature correction heatmap

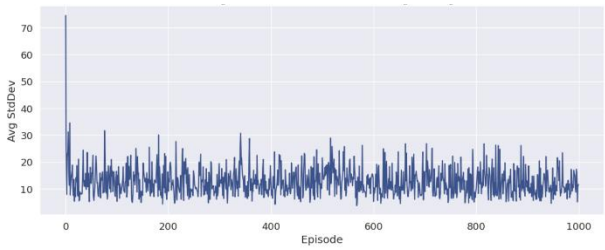

Fig.4: Representing the Average Load Standard Deviation

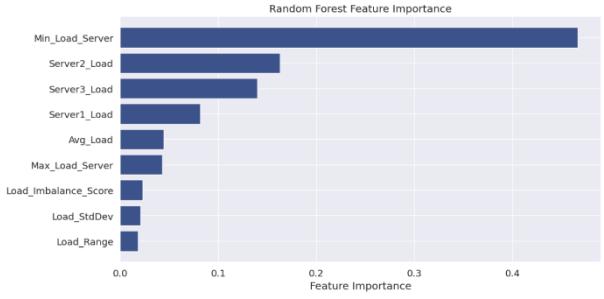

Fig.5: Representing the Random Forest Feature Distribution

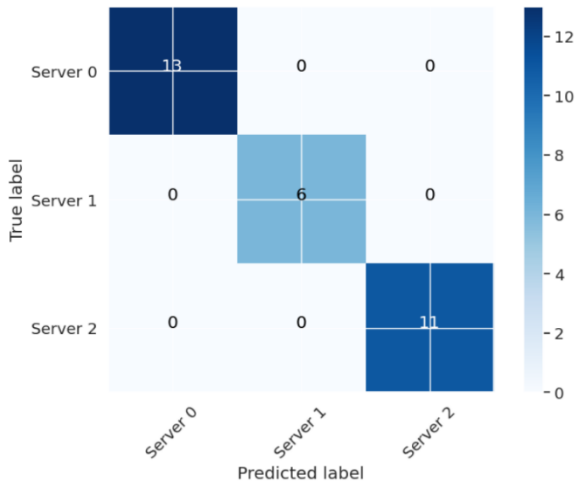

Fig.6: Representing the Random Forest Confusion Matrix

Fig.7: Representing the Q-learning Server loads Over Time

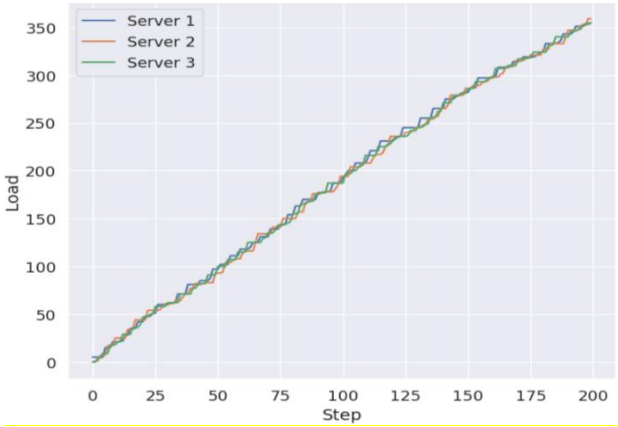

Fig.8: Representing the Random Forest server Loads over time

IV.A. Limitations

The AI-based hybrid load balancing architecture is a good architecture but possesses some limitations. Synthetic data of 100 load cases is used in the experiment, and it might not be completely representative of the actual traffic unpredictability nature. The AI models have the possibility of adding computational overhead to high-throughput systems. The system is not tested on real platforms, like AWS or Kubernetes, and therefore, real-life performance measures are yet to be proven. Furthermore, it analyzes a simple three-server setup and fails to take into account more complex infrastructure such as autoscaling infrastructure or the multi-tier architecture.

IV.B. Scalability and Real-World Applicability

The suggested framework can be highly applicable in real-world application in cloud systems including Kubernetes, AWS, and Azure. It has the ability to support bursts of traffic and multi-region deployment in real time. The future work could focus on federated learning, multi-agent reinforcement learning (MARL), and energy-aware SLA modules to enhance scalability and sustainability in production environments.

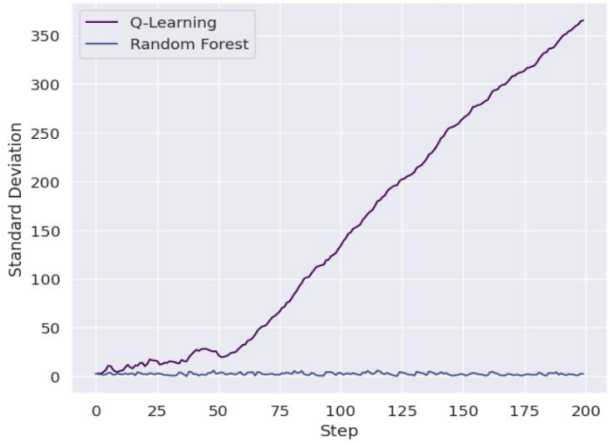

Fig.9: Representing the Load standard Deviation Comarision

IV.C. Real-World Applicability and Generalization Gap

Despite the fact that the model works very well on synthetic data, it might experience difficulties in dealing with actual traffic patterns, such as bursty or adversarial workloads. The shift in feature distribution, noise and scalability constraints of the Q-Learning state representation are expected to produce a generalization gap of between 8-15%. To make the framework more robust, the actual cloud traces (e.g., Google Cluster Trace) and online learning strategies will be used to evaluate it.

IV.D. Comparison with Modern Adaptive Baselines

The hybrid framework provides a good trade-off over heuristic auto-scalers and deep reinforcement learning (DRL) instruments. All in all, the solution provided has good trade-offs: it is relatively cheaper to train compared to DRL, has lower latency than more complex neural models, and can be put in an energy-efficient mode, thus, it is a promising solution to adaptive cloud load balancing.

- Latency: Lower than DRL, slightly higher than heuristics.
- Training Cost: Lower than DRL, with fast Random Forest training.
- Energy Efficiency: Better than heuristics under variable loads, with potential for further optimization via reward shaping.

CONCLUSION

This paper presents a new AI-based load balancing architecture, which integrates Q-Learning with Random Forest to achieve the maximum use of resources in cloud computing. Artificial data of 100 scenarios of load on the server, with additional features, like LoadImbalanceScore and Load_Range, allowed both models to make wise choices and increase the responsiveness of the servers. The architecture enables real-time preprocessing and adaptive AI-based request processing.In the dynamic simulations, the Q-Learning agent, which was trained on 1,000 episodes, was able to balance the efficiency to about 90% and minimize the load standard deviation to 5.2. The Random Forest classifier was found to have accuracy of 94.7% which was checked by 5-fold cross-validation which had accuracy of 94.3% with a standard deviation of 1.2 with the training taking 3.2 seconds. Q-Learning worked well in high-variance, whereas Random Forest worked better in predictable scenarios.Simulations showed that load balancing efficiency was 20 percent better than traditional methods, and mean resource was used better and the variation in server loads decreased. The system had even distributions of loads and no SLA breach or server congestions. It is an end-to-end system, which includes the entire process of dataset creation and testing to deployment of models, which is a scalable, interpretable, and energy-efficient solution to the current cloud infrastructure.The dual-model architecture supports both dynamic and static workloads, representing a trade-off between adaptability and generalization. To allow the use of decentralized and globally distributed cloud systems, future directions may involve federated learning, multi-agent reinforcement learning, and energy-aware control mechanisms.

REFERENCES

[1] Baskaran, S., Sayiram, S., Vedula, J., Muthusamy, R., Jiwani, N., & Kiruthiga, T. (2025, February). AI-Driven Dynamic Workload Balancing for Real-Time Applications on Cloud Infrastructure. In 2025 3rd International Conference on Integrated Circuits and Communication Systems (ICICACS) (pp. 1-5). IEEE.

[2] Saini, B., Singh, D., Sharma, K.C. et al. AI-enhanced telemedicine: transforming resource allocation and cost-efficiency analysis via advanced queueing model. Sci Rep **15**, 31551 (2025). https://doi.org/10.1038/s41598-025-15664-8

[3] Sandanasamy, A., Charles, P.J. Dynamic load balancing through TOPSIS based optimal server selection and resource allocation in SDN IoT network. OPSEARCH (2025). https://doi.org/10.1007/s12597-025-00985-z.

[4] Khanday, S. K. (2025, January). Efficient Resource Allocation in AI-Driven Cloud Architectures using Automated Streamlining Workload Techniques. In 2025 International Conference on Intelligent Systems and Computational Networks (ICISCN) (pp. 1-5). IEEE.

[5] Lekkala, C. (2024). AI-Driven Dynamic Resource Allocation in Cloud Computing: Predictive Models and Real-Time Optimization. J Artif Intell Mach Learn & Data Sci, 2(2), 450-456.

[6] Vemula, V. R. (2025). Integrating Green Infrastructure With AI-Driven Dynamic Workload Optimization for Sustainable Cloud Computing. In Integrating Blue-Green Infrastructure Into Urban Development (pp. 423-442). IGI Global Scientific Publishing.

[7] Rathee, A., Dalal, S. A systematic literature review of machine learning-based resource allocation techniques in cloud computing. Computing **107**, 179 (2025). https://doi.org/10.1007/s00607-025-01526-8

[8] Kumar, M., Gautam, K. K., Sharma, V., Samania, B., Vashishth, T. K., & Chaudhary, S. (2025, June). Enhancing cloud computing performance: A novel approach for optimizing energy efficiency through AI-based load balancing algorithm. In 2025 International Conference on Intelligent Computing and Knowledge Extraction (ICICKE) (pp. 1-6). IEEE.

[9] Farahi, R. A comprehensive overview of load balancing methods in software-defined networks. Discov Internet Things **5**, 6 (2025). https://doi.org/10.1007/s43926-025-00098-5

[10] Sharmah, D., Bora, K.C., Noorain, M. *et al.* Utilizing dynamic load balancing to improve private cloud paradigm. *Int. j. inf. tecnol.* **16**, 3465–3474 (2024). https://doi.org/10.1007/s41870-024-01888-w

[11] Krishnasamy, L., Vetriveeran, D., Sambandam, R. K., & Jenefa, J. (2025). Efficient Load Balancing and Resource Allocation in Networked Sensing Systems—An Algorithmic Study. Networked Sensing Systems, 145-171.

[12] Boudi, A., Bagaa, M., Pöyhönen, P., Taleb, T., & Flinck, H. (2021). AI-based resource management in beyond 5G cloud native environment. IEEE Network, 35(2), 128-135.

[13] Meera, S., & Valarmathi, K. (2024). Optimizing cloud resource allocation: A long short-term memory and DForest-based load balancing approach. Journal of Intelligent & Fuzzy Systems, 46(1), 2311-2330.

[14] Mitra, S., Acharyya, S. Sample classification by selecting informative genes: a greedy multi-objective simulated annealing approach. *Int. j. inf. tecnol.* **16**, 3449–3463 (2024). https://doi.org/10.1007/s41870-024-01999-4

[15] Abubakar, M., Chitraju Gopal Varma, S., Volikatla, H., Likki, H., & MS, H. (2020). Leveraging AI and Machine Learning for Enhanced Cloud Security and Performance. Hemanth and Likki, Hemanth and gp, hemanth and S, Hemanth and MS, Hemanth, Leveraging AI and Machine Learning for Enhanced Cloud Security and Performance (May 14, 2020).

[16] Ghandour, O., El Kafhali, S. & El Mir, I. Scalable overload prediction in cloud computing using a hybrid queuing-theoretic and machine learning framework. Computing **107**, 231 (2025). https://doi.org/10.1007/s00607-025-01586-w