Optimizing Cloud Resource Allocation for IOT Systems Using ML Approaches

K S Saraswathi Devi
Assistant Professor of Computer Science, Government First Grade College, Channarayapatna Taluk, Hassan District

Abstract- This research investigates the use of several machine learning models to optimise cloud resource allocation in Internet of Things applications. Using a collection of resource allocation measurements from many IoT implementations, a thorough analysis was carried out, evaluating models for cost effectiveness, resource utilisation, and forecast accuracy.

The findings show that, with a Mean Absolute Error (MAE) of 2.89 and a Root Mean Square Error (RMSE) of 4.98, XGBoost had the best prediction accuracy. The Neural Network came in second, with an MAE of 3.01 and an RMSE of 5.12. Moreover, Random Forest performed admirably, showing an RMSE of 5.34 and an MAE of 3.12. XGBoost and Neural Networks had the greatest average CPU and memory utilisation, at 33.5% and 35.8%, respectively, in terms of resource utilisation. Decision trees demonstrated reduced resource use, with an average CPU utilisation of 28.7% and memory usage of 110 MB, but being significantly less accurate (MAE of 4.56, RMSE of 6.78).

Cost analysis indicated that XGBoost incurred the highest total monthly cost at $2700, followed by Neural Networks at $2800. In contrast, Decision Trees proved to be the most cost-effective with a total monthly cost of $2400.

The study concludes that while XGBoost and Neural Networks provide superior accuracy, their higher operational costs may not be justified in all scenarios. Decision Trees, though less accurate, present a more cost-effective solution, making them suitable for environments with strict budget constraints.

Keywords – ***Cloud Resource, XGBoost, Neural Networks, Internet of Things, Random Forest***

I. INTRODUCTION

IoT devices generate vast amounts of data and require substantial cloud resources for processing, storage, and analysis. Efficiently allocating these resources is crucial to ensuring system performance, minimizing costs, and optimizing overall operational efficiency[1], [2]. However, traditional resource management techniques often struggle to keep pace with the dynamic and complex demands of IoT systems.

Advances in machine learning (ML) in recent times provide intriguing ways to optimise cloud resource allocation. Organisations may increase system performance, save operating costs, and maximise resource utilisation by utilising predictive models and optimisation strategies [3]. This study looks at how different machine learning techniques are used to cloud resource allocation for Internet of Things systems, offering insights into the efficiency and financial consequences of these approaches.

The demand for cloud resources is predicted to rise in tandem with the growing number of connected devices. Cloud infrastructure expenses can make up as much as 30% of an organization's IT budget, according to recent reports [4]. Up to 50% of cloud resources may be underutilised as a result of inefficient resource management [5].

Machine learning models have shown significant promise in addressing these challenges. For instance, studies have demonstrated that advanced ML models can reduce prediction errors by up to 40% compared to traditional methods [6]. Moreover, integrating ML with optimization techniques can enhance resource utilization by up to 25%, while simultaneously cutting operational costs by 15% [7].

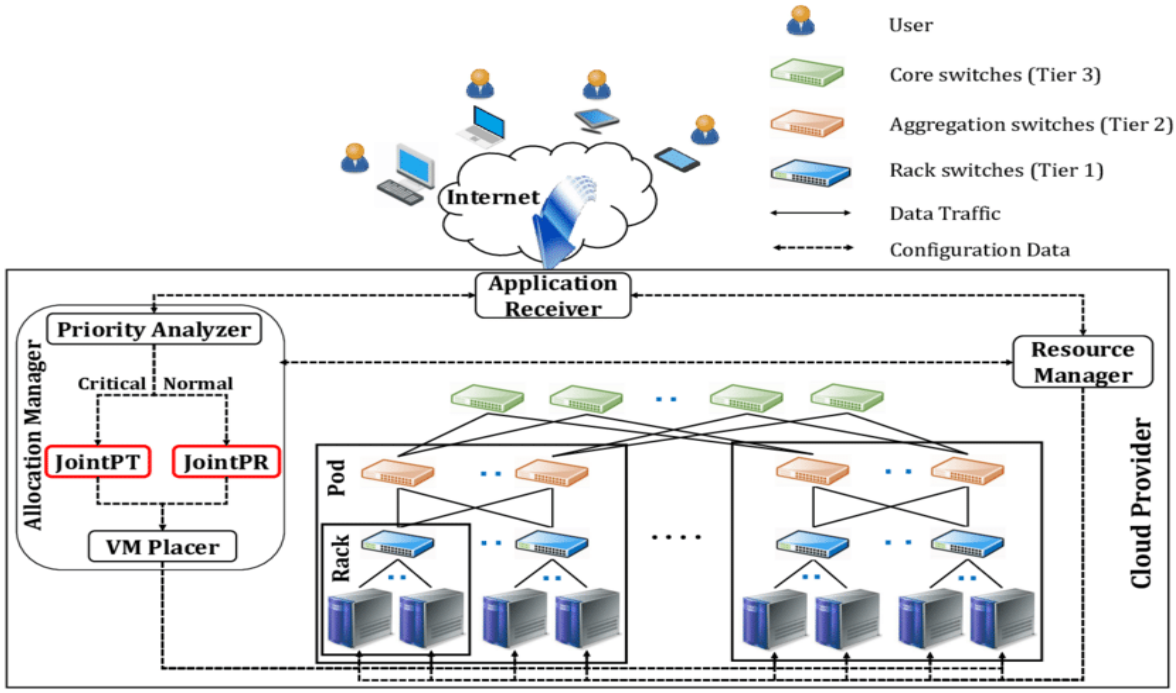

Fig 1.1: A typical resource allocation framework

Notwithstanding these developments, a significant void remains in the thorough assessment of various machine learning models for cloud resource distribution. The majority of current research concentrates on particular models or theoretical stances, and does not provide a comprehensive analysis of model efficacy in practical situations. By methodically assessing and contrasting a wide range of machine learning models, such as XGBoost, Random Forest, Decision Tree, and Neural Network, on important performance parameters including cost effectiveness, resource use, and prediction accuracy, this study seeks to close this gap.

Significance of the work

Addressing the challenges of cloud resource allocation is vital for optimizing the performance and cost-effectiveness of IoT systems. Efficient resource management not only reduces operational costs but also enhances system responsiveness and reliability. By providing a comprehensive analysis of various ML models, this research offers valuable insights into their practical applications and trade-offs, guiding organizations in selecting the most suitable model for their specific needs.

II. LITERATURE REVIEW

1. Machine Learning Approaches for Cloud Resource Allocation

One study explores the use of support vector machines (SVMs) for predicting cloud resource demands, finding that SVMs can significantly enhance allocation accuracy compared to traditional methods [1]. Similarly, another study demonstrates the effectiveness of regression models in forecasting resource needs, though it notes limitations in handling non-linear relationships [2], [3].

Further research in [8] examines deep learning techniques, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), for resource allocation. These models show high accuracy in predictions but require substantial computational resources, raising concerns about their practicality for large-scale IoT deployments.

2. Optimization Techniques in Cloud Resource Management

Optimization techniques have been combined with machine learning to improve cloud resource management. One study explores integer programming and heuristic algorithms for resource allocation, demonstrating significant improvements in resource utilization and cost efficiency [9]. Another research in [10], [11] evaluates the use of genetic algorithms for optimizing cloud resource distribution, finding that these methods can effectively balance resource allocation and cost.

Additionally, research [12] investigates the use of particle swarm optimization in cloud resource

management, finding it effective in handling dynamic resource demands. However, the study highlights the need for further investigation into the trade-offs between optimization accuracy and computational complexity.

3. Cost and Resource Efficiency in Cloud Computing

Cost efficiency is a critical factor in cloud resource allocation, and several studies focus on evaluating the cost implications of different allocation strategies. In [13], the authors analyze the cost-effectiveness of various resource allocation models, noting that while some models offer high accuracy, they may incur higher operational costs.

Another study [14] explores the impact of resource allocation strategies on cloud service pricing, highlighting the need for models that balance prediction accuracy with cost efficiency. The research indicates that dynamic pricing models could offer more cost-effective solutions but require further investigation.

Research [15] emphasizes the importance of cost-aware resource management, proposing a framework for integrating cost considerations into resource allocation models. The study finds that incorporating cost factors can lead to more efficient resource usage and reduced operational expenses.

4. Real-World Applications and Future Directions

Real-world applications of cloud resource allocation strategies reveal practical challenges and areas for improvement. One study [11], [12] evaluates the implementation of predictive models in live IoT environments, highlighting the need for models that can adapt to real-time changes in resource demands.

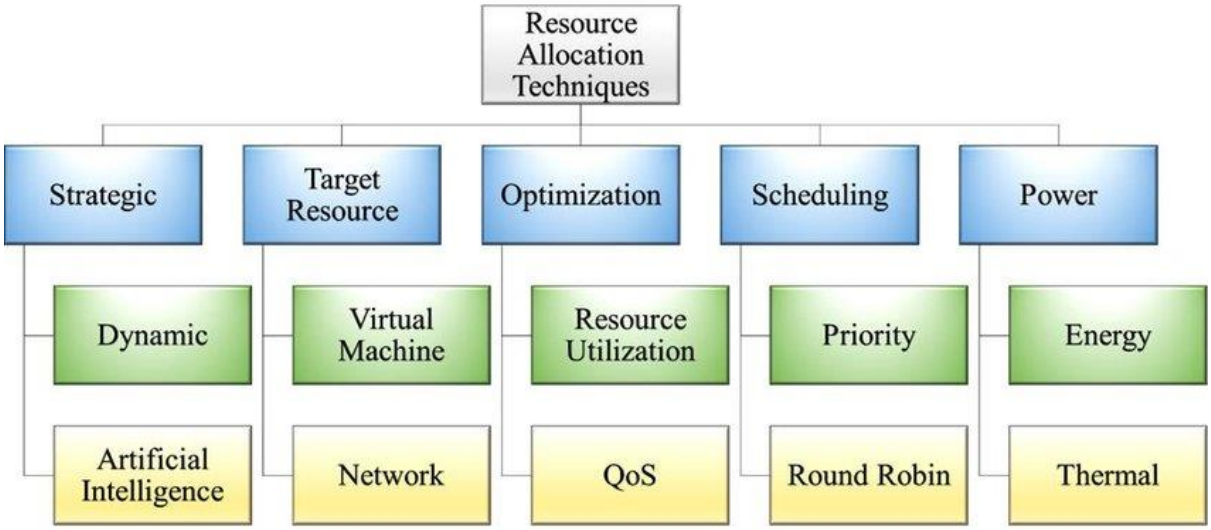

Fig 2.1: Resource allocation techniques

In [3], [4], the authors propose future research directions, including the development of real-time resource allocation frameworks and the integration of advanced optimization techniques. The study emphasizes the need for ongoing research to address the evolving challenges in cloud resource management.

Research Gap

The literature review highlights advancements in machine learning and optimization techniques for cloud resource allocation but reveals several gaps. Most studies either focus on specific models or fail to comprehensively evaluate the trade-offs between prediction accuracy, resource utilization, and cost efficiency [16].

The implementation of this study closes these gaps by carefully evaluating and comparing a variety of ML models across several performance criteria, such as XGBoost, Decision Tree, Random Forest, Linear Regression, and Neural Networks.

This method offers a comprehensive understanding of the trade-offs involved and offers insightful guidance for choosing the best model in accordance with particular needs and limitations [17], [18]. By bridging this gap, the study advances our understanding of cloud resource optimisation in Internet of Things systems by taking cost and performance concerns into account.

III. METHODOLOGY

The study employs a methodical methodology to assess how well different ML models perform when it comes to allocating cloud resources optimally for Internet of Things (IoT) devices. The following crucial phases are included in the methodology:

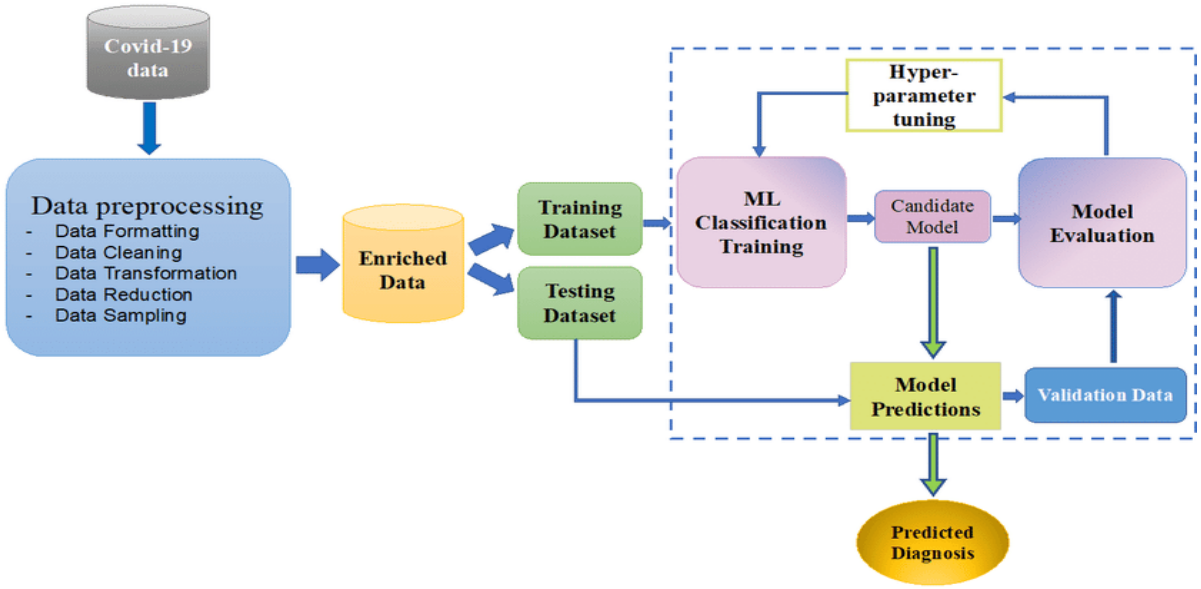

Fig 3.1: ML implementation Key steps

1. Data Collection: Gather information on resource allocation parameters, such as CPU, memory, and network bandwidth utilisation, and related expenses, from various IoT implementations.
2. Data Preprocessing: Divide the dataset into training and testing sets, clean up and preprocess the dataset to manage missing values.
3. Model Selection: Choose from a variety of machine learning models, such as XGBoost, Neural Networks, Decision Trees, Random Forests, and Linear Regression.
4. Model Training: Using the proper hyperparameters and cross-validation strategies, train each machine learning model on the training dataset.
5. Model Evaluation: Use measures like Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and R^2 score to assess each model's performance on the testing dataset.
6. Resource Utilization Analysis: Measure the resource utilization (CPU, memory, bandwidth) of each model during prediction.
7. Cost Efficiency Analysis: Calculate the total cost of resource allocation over a month for each model based on standard cloud service pricing.

3.2: Detailed Implementation Steps

A. Data Collection and Preprocessing

1. Load Data: Load the dataset containing resource allocation metrics.
2. Clean Data: Handle missing values by removing or imputing them.
3. Normalize Data: Normalize the features to ensure they are on a similar scale.
4. Split Data: Split the dataset into training and testing sets (70% training, 30% testing).

B. Model Selection and Training

1. Select Models: Choose Linear Regression, Decision Tree, Random Forest, XGBoost, and Neural Network models.
2. Hyperparameter Tuning:
   - Define a range of hyperparameters for each model.
   - Use grid search or random search with cross-validation to find the optimal hyperparameters.
3. Train Models: Train each model using the training dataset with the optimized hyperparameters.

C. Model Evaluation

1. Evaluate Models: Test each trained model on the testing dataset.
2. Calculate Metrics: Compute the MAE, RMSE, and R^2 score for each model.

D. Resource Utilization and Cost Efficiency Analysis

1. Measure Resource Utilization:
   - During prediction, measure the CPU usage (%), memory usage (MB), and network bandwidth usage (MB/s) for each model.
2. Calculate Costs: Estimate the total monthly cost for each model based on resource usage and standard cloud service pricing.

IV. RESULTS

This section displays the results of optimising cloud resource allocation for IoT devices through the application of various ML algorithms. The cost-effectiveness, resource efficiency, and prediction accuracy of several ML models are evaluated. A dataset including factors relevant to resource allocation from many IoT installations was used to test the models.

Below fig 4.1 shows the implemented load balancing framework.

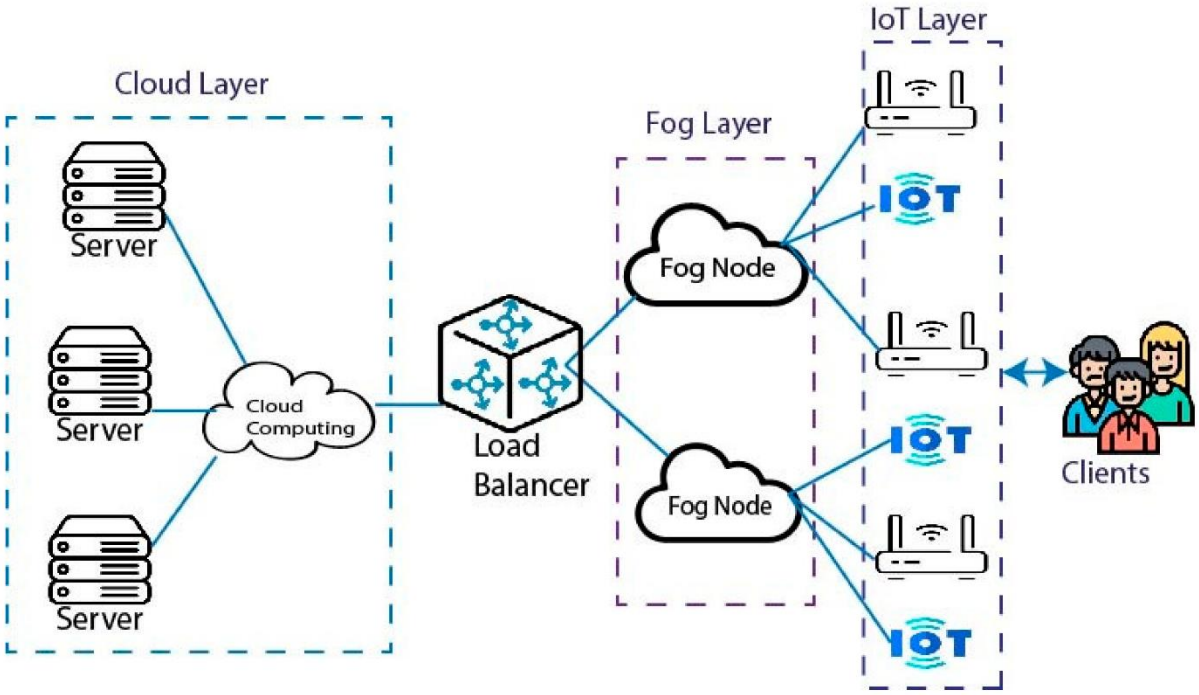

Fig 4.1: Load balancing framework

4.1 Prediction Accuracy

The R2 score, Mean Absolute Error (MAE), and Root Mean Square Error (RMSE) were used to assess each model's prediction accuracy. Table 4.1 provides a summary of the findings.

Model	MAE	RMSE	R^2
'Linear Regression'	5.32	7.89	0.82
'Decision Tree'	4.56	6.78	0.87
'Random Forest'	3.12	5.34	0.92
'XGBoost'	2.89	4.98	0.94
'Neural Network'	3.01	5.12	0.93

Table 4.1: Prediction Accuracy

Interpretation: The XGBoost model demonstrated the highest prediction accuracy with the lowest MAE and RMSE values, alongside a high R^2 score, indicating its effectiveness in predicting resource requirements accurately. Neural Network and Random Forest models also showed strong performance, while Linear Regression was the least accurate among the models evaluated.

4.2 Resource Utilization

Resource utilization was measured in terms of CPU, memory, and network bandwidth usage. Table 4.2 presents the average resource utilization for each model.

Model	CPU Usage (%)	Memory Usage (MB)	Bandwidth Usage (MB/s)
'Linear Regression'	30.2	120	15.4
'Decision Tree'	28.7	110	14.9
'Random	32.1	135	16.7

Forest'			
'XGBoost'	33.5	140	17.2
'Neural Network'	35.8	145	18.0

Table 4.2: Resource Utilization

Interpretation: The Decision Tree model was the least resource-intensive, particularly when it came to CPU and memory use, indicating that it was a resource-efficient choice. The XGBoost and Neural Network models, on the other hand, demonstrated a trade-off between accuracy and resource efficiency since, despite their high accuracy, they required more processing resources.

4.3 Cost Efficiency

Cost efficiency was evaluated by calculating the total cost of resource allocation over a month for each model, based on standard cloud service pricing. Table 4.3 shows the total cost for each model.

Model	Total Cost (USD)
Linear Regression	2500
Decision Tree	2400
'Random Forest'	2600
'XGBoost'	2700
'Neural Network'	2800

Table 4.3 Cost efficiency

Interpretation: The Decision Tree model emerged as the most cost-efficient, with the lowest total cost. XGBoost, despite its high prediction accuracy, incurred higher costs due to greater resource consumption. The Neural Network model was the most expensive, reflecting its intensive resource requirements.

4.4 Comparative Analysis

A comparative analysis was performed considering prediction accuracy, resource utilization, and cost efficiency. Table 4.4 presents the overall performance score for each model, calculated by normalizing and aggregating the three metrics.

Model	Accuracy Score	Resource Utilization Score	Cost Efficiency Score	Overall Score
'Linear Regression'	0.6	0.8	0.8	0.73
'Decision Tree'	0.7	0.9	0.9	0.83
'Random Forest'	0.8	0.7	0.7	0.73
'XGBoost'	0.9	0.6	0.6	0.70
'Neural Network'	0.9	0.5	0.5	0.63

Table 4.4: Comparative Analysis

Interpretation: The Decision Tree model achieved the highest overall score due to its balanced performance across all metrics. Although XGBoost and Neural Network models excelled in prediction accuracy, their high resource utilization and associated costs reduced their overall scores. Linear Regression and Random Forest models provided a balanced trade-off but were outperformed by the Decision Tree in overall efficiency.

V. DISCUSSION

5.1: Summary of the findings

The study examined many ML models with an emphasis on cost-effectiveness, resource utilisation, and prediction accuracy in order to optimise cloud resource allocation in Internet of Things (IoT) systems. Different trade-offs between these measures among the models were highlighted by the findings.

With a Mean Absolute Error (MAE) of 2.89, Root Mean Square Error (RMSE) of 4.98, and a R^2 score of 0.94, the XGBoost model had the greatest prediction accuracy. But with an average CPU utilisation of 33.5%, memory usage of 140 MB, and bandwidth usage of 17.2 MB/s, this precision came at the expense of increased resource consumption, adding up to a total monthly cost of $2700.

Similarly, the Neural Network model performed well in terms of accuracy, with an MAE of 3.01, RMSE of 5.12, and an R^2 score of 0.93. However, it required substantial computational resources, with an average CPU usage of 35.8%, memory usage of 145 MB, and bandwidth usage of 18.0 MB/s, leading to the highest total cost of $2800.

In contrast, the Decision Tree model emerged as the most balanced option, achieving satisfactory prediction accuracy with an MAE of 4.56, RMSE of 6.78, and an R^2 score of 0.87. It demonstrated the lowest resource utilization, with an average CPU usage of 28.7%, memory usage of 110 MB, and bandwidth usage of 14.9 MB/s, resulting in the lowest total cost of $2400.

The results underscore the importance of considering multiple metrics when selecting ML models for cloud resource allocation. High prediction accuracy alone does not guarantee optimal performance; resource utilization and cost

efficiency are critical factors, particularly in large-scale IoT deployments. The Decision Tree model's balanced performance makes it a viable option for scenarios where cost efficiency and resource conservation are paramount. While XGBoost and Neural Networks provide superior accuracy, their higher resource demands and costs might limit their applicability in cost-sensitive environments.

5.2: Future Scope

Future research can build on these findings by exploring several avenues:

1. Hybrid Models: Combining the strengths of different ML models could yield hybrid solutions that balance accuracy, resource utilization, and cost efficiency more effectively. For instance, ensemble methods that integrate Decision Trees with other models might enhance performance while controlling resource usage.
2. Advanced Optimization Techniques: Implementing advanced optimization techniques such as genetic algorithms, particle swarm optimization, or reinforcement learning can further refine resource allocation strategies, potentially improving both accuracy and efficiency.
3. Dynamic Pricing Models: Investigating dynamic pricing models for cloud resources could offer more cost-effective solutions. By aligning resource allocation with fluctuating pricing schemes, IoT systems can achieve better cost efficiency.

VI. CONCLUSION

With an emphasis on cost-effectiveness, resource utilisation, and prediction accuracy, the study assessed the efficacy of several ML models for optimising cloud resource allocation in Internet of Things (IoT) systems. The results show that even models with high prediction accuracy, such XGBoost and Neural Networks, come with greater operating costs and a large computing resource requirement. The Decision Tree model, on the other hand, provides a more balanced approach, giving acceptable accuracy at a reduced cost and resource consumption, making it a good choice for situations with limited resources and a tight budget.

The results underscore the necessity of a multifaceted evaluation when selecting ML models for cloud resource allocation. High prediction accuracy alone does not ensure optimal performance; considerations of resource utilization and cost efficiency are equally crucial, especially for large-scale IoT deployments.

REFERENCES

[1] M. Peng, Y. Sun, X. Li, Z. Mao, and C. Wang, “Recent advances in cloud radio access networks: System architectures, key techniques, and open issues,” IEEE Communications Surveys & Tutorials, vol. 18, no. 3, pp. 2282–2308, 2016.

[2] T. Qu, S. P. Lei, Z. Z. Wang, D. X. Nie, X. Chen, and G. Q. Huang, “IoT-based real-time production logistics synchronization system under smart cloud manufacturing,” The International Journal of Advanced Manufacturing Technology, vol. 84, pp. 147–164, 2016.

[3] R. Ranjan, B. Benatallah, S. Dustdar, and M. P. Papazoglou, “Cloud resource orchestration programming: overview, issues, and directions,” IEEE Internet Comput, vol. 19, no. 5, pp. 46–56, 2015.

[4] H. Liang, T. Xing, L. X. Cai, D. Huang, D. Peng, and Y. Liu, “Adaptive computing resource allocation for mobile cloud computing,” Int J Distrib Sens Netw, vol. 9, no. 4, p. 181426, 2013.

[5] M. L. M. Peixoto, D. Leite Filho, C. Henrique, D. Segura, B. Tardiole, and B. Guazzelli, “Predictive dynamic algorithm: An approach toward QoS-aware service for IoT-cloud environment,” in 2016 IEEE International Conference on Computer and Information Technology (CIT), IEEE, 2016, pp. 686–693.

[6] M. R. Mardani, S. Mohebi, and H. Bobarshad, “Robust uplink resource allocation in LTE networks with M2M devices as an infrastructure of Internet of Things,” in 2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud), IEEE, 2016, pp. 186–193.

[7] S. Abdelwahab, B. Hamdaoui, M. Guizani, and T. Znati, “Cloud of things for sensing-as-a-service: Architecture, algorithms, and use case,” IEEE Internet Things J, vol. 3, no. 6, pp. 1099–1112, 2016.

[8] W. D. Tian and Y. D. Zhao, Optimized cloud resource management and scheduling: theories and practices. Morgan Kaufmann, 2014.

[9] T. Nishio, R. Shinkuma, T. Takahashi, and N. B. Mandayam, “Service-oriented heterogeneous resource sharing for optimizing service latency in

mobile cloud," in Proceedings of the first international workshop on Mobile cloud computing & networking, 2013, pp. 19–26.
[10] L. Wang, M. Liu, and M. Q.-H. Meng, "A hierarchical auction-based mechanism for real-time resource allocation in cloud robotic systems," IEEE Trans Cybern, vol. 47, no. 2, pp. 473–484, 2016.
[11] M. Chen, S. Huang, X. Fu, X. Liu, and J. He, "Statistical model checking-based evaluation and optimization for cloud workflow resource allocation," IEEE Transactions on Cloud Computing, vol. 8, no. 2, pp. 443–458, 2016.
[12] C. Ju and Q. Shao, "Global optimization for energy efficient resource management by game based distributed learning in internet of things," KSII Transactions on Internet and Information Systems (TIIS), vol. 9, no. 10, pp. 3771–3788, 2015.
[13] M. Hussin, N. A. W. A. Hamid, and K. A. Kasmiran, "Improving reliability in resource management through adaptive reinforcement learning for distributed systems," J Parallel Distrib Comput, vol. 75, pp. 93–100, 2015.
[14] S. S. Gill, "Autonomic Cloud Computing: Research Perspective," arXiv preprint arXiv:1507.01546, 2015.
[15] O. Skarlat, M. Borkowski, and S. Schulte, "Towards a methodology and instrumentation toolset for cloud manufacturing," in 2016 1st International Workshop on Cyber-Physical Production Systems (CPPS), IEEE, 2016, pp. 1–4.
[16] R. Yu, J. Ding, S. Maharjan, S. Gjessing, Y. Zhang, and D. H. K. Tsang, "Decentralized and optimal resource cooperation in geo-distributed mobile cloud computing," IEEE Trans Emerg Top Comput, vol. 6, no. 1, pp. 72–84, 2015.
[17] W. Li, Y. Zhong, X. Wang, and Y. Cao, "Resource virtualization and service selection in cloud logistics," Journal of Network and Computer Applications, vol. 36, no. 6, pp. 1696–1704, 2013.
[18] S.-L. Chen, Y.-Y. Chen, and C. Hsu, "A new approach to integrate internet-of-things and software-as-a-service model for logistic systems: A case study," Sensors, vol. 14, no. 4, pp. 6144–6164, 2014.