ONLINE ISSN: 3007-3197 | PRINT ISSN: 3007-3189

http://amresearchreview.com/index.php/Journal/about

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 3, Issue 4 (2025)

Intelligent Resource Allocation in Cloud Computing Environments: Leveraging Machine Learning for Dynamic Workload Balancing, Cost Efficiency, and Performance Optimization

[1]Faisal Haroon

Article Details

Keywords: Cloud Computing, Resource Allocation, Machine Learning, Deep Q-Learning, Random Forest, SLA Compliance, Autoscaling, Cost Optimization, Workload Prediction, CloudSim.

Faisal Haroon
IT Consultant, Comsats University, Abbottabad Campus.
faisalharoon_4@yahoo.com

ABSTRACT

Cost optimization is widely considered a critical aspect of resource management in current cloud computing scenarios to consider service quality, resource utilization and SLAs. The traditional static or the rule based algorithms are not well suited to handle the dynamics, heterogeneity and variability of workloads usually found in cloud data centers of vast sizes. To address these challenges, the current research introduces an intelligent resource allocation framework that uses machine learning. In particular, the workload prediction is achieved with a Random Forest model, while the resource management, scheduling, and auto scaling is done by a Deep Q-Learning agent. The proposed framework was tested with CloudSim Plus simulator and Google Cluster Trace dataset comparing its performance with FCFS, Round Robin and threshold-based methods. It is seen that the resource utilization increases up to 85.3%, SLA violation is reduced to 4.1% and the cost of execution is reduced by 28% which is based on conventional strategies where the throughput of the tasks achieved is 674 tasks per hour. The results demonstrate how machine learning can help build smarter, efficient, and cost-aware cloud applications and infrastructure to support responsive and autonomously optimizing cloud systems.

INTRODUCTION

Cloud computing has emerged as a “revolutionary” phenomenon in the field of computing since it provides scalable, flexible and affordable solutions through resource provisioning. Computing, storage and networking services that do not require individuals and companies to make capital investments in IT hardware assure a broad variety of uses, including web hosting, scientific modeling, storage of massive amounts of data, and artificial intelligence (Mell & Grance, 2011). The elasticity and the multi-tenancy of the cloud platforms therefore poses a major problem of resource management. Efficient utilization of these resources is desirable for QoS provision, reduction of operational costs while at the same time meeting stringent SLAs (Buyya et al., 2009).

Current techniques in resource management mainly incorporate techniques like fixed thresholding, rule based scaling and simple scheduling algorithms (Beloglazov and Buyya, 2012). These methods are simple and do not consume much CPU cycles, but are not effective for the dynamic and diverse workloads of the cloud computing systems. Static rules are not dynamic enough to work at real-time; therefore, this issue leads to over-provisioning, making the costs high or under-provisioning, which results in poor performance and a breach of the agreed service level (Calheiros et al., 2011). Additionally, with the introduction of containers, microservices, and serverless services the nature of workloads varies and splits into a far larger number of smaller parts, which requires even more sophisticated approaches.

In order to overcome these limitations, there is an approach toward utilizing machine learning (ML) as an efficient solution for intelligent resource management in a cloud computing environment. With the use of ML, systems can analyze prior data, anticipate future usage of resources, and independently make allocation decisions (Xu et al., 2020). ML models are different from static algorithms in the sense that they can easily identify nonlinearity present in associations between a large number of metrics such as CPU usage, memory consumption, and network bandwidth. For example, for prediction of resource usage, regression based models have been used and for the purpose of workload grouping, clustering methodologies are used (Qiu et al., 2016; Zhang et al., 2020). Reinforcement learning (RL) strategies have also been employed as the agents learn the best resource allocation strategies through trial and error within the environment whereby the performance of the system is either rewarded positively or punished (Mao et al., 2016).

Recent developments in the field of deep learning took the existing possibilities of the use of ML for cloud resource control to a new level. DNNs, LSTM, and CNNs have the ability to cater the workload prediction of time-series and perform spatial-temporal resource examination (Tang et al., 2020; Chen et al., 2019). When connected with cloud orchestrations such as Kubernetes or OpenStack, these models can proactively control resources consisting of VMs, containers, and services based on the current application traffic. Besides, integrating ML with control theory, metaheuristic, or fog-edge computing models have also been proposed to solve MOOPs in large distributed systems (Wang et al., 2021).

However, the following issues are still issues; Some important questions that arise are the ability to generalize the ML models across all forms of clouds and workloads that can be unpredictably distributed. In one circumstance, a particular model may not be as effective in

another due to variations in peculiarities, in terms of the available hardware, flow volumes, or the specifications of the applications (Islam et al., 2012). Another issue is the training and deployment times that need to be put into the models and the other resources that are used to train them that may counterbalance the efficiency gains on the usage of resources. Also, there are challenges in terms of data privacy and security when training models from user level telemetry data, particularly when the case is multi-tenant public cloud (Liu et al., 2021).

Therefore, the purpose of this paper is to discuss a novel machine learning based intelligent resource allocation framework for cloud computing. This paper examines the use of supervised learning and deep reinforcement learning to improve workload distribution, cost optimization and system performance. In simulation-based experiments, the presented approach is compared with other resource management techniques in a real-world experimental setup consisting of three real datasets.

This paper also attempts to blur the line between prediction and decision-making in order to allow a scalable, adaptive, and cost-efficient resource management to fit the next generation cloud environments.

LITERATURE REVIEW

Resource management in cloud computing has been a major issue of concern over the past decade given that resource management is a critical determinant on the achievement of efficient utilization of Virtualized resources while adhering to contractual service level agreements. In the past, the approaches used to manage the resources in the cloud systems were Rules based or the threshold mechanisms. Despite their simplicity and low computational costs, they are less effective in such dynamic contexts where workloads are stochastic and volatile as has been pointed out in Kliazovich et al. (2013). In recent years, therefore, there has been a shift towards intelligent and autonomous approaches to the field where ML is well equipped to play a significant role.

In the initial years of the implementation of ML techniques in the domain of cloud computing, research investigations were mainly centered on the use of the approach to predict the demand for IT resources. For example, Lama and Zhou (2012) discussed a framework of predicting resource provisioning through a past workload and an adaptive provisioning scheme based on the queuing theory and linear regression. Similarly, Yazdanov and Fetzer (2013) used time series analysis for the proactive computation of VM provisioning that aimed at preventing

SLA violations. These models were precursors to more complex ML-based techniques yet were weak, rigid, and linear in the sense that they could only handle simple patterns of workloads.

When the concept of workload classification started emerging, classification techniques evolved and later clustering technologies became stronger to support the consolidation of workloads. Paya, and Marinescu in 2014 integrated a workload classification model that utilized support vector machine (SVM) for a better decision making on scale-out on IaaS clouds. In the same case, Goudarzi et al. (2015) used the k-means clustering technique to easily cluster similar workloads for the purpose of resource pooling. These approaches considerably cut down wastage of resources through making sure that the VMs that were likely to consume resources in the same manner were assigned to the same physical resources.

In recent times, deep learning techniques have been explored in view of their capability to address the scatters and concealed patterns in the use of resources. Al-Dhuraibi et al. (2018) applied deep belief networks in auto-scaling of containerized application on cloud through showing that it outperforms shallow models in terms of accurate predictions. Moreover, Zhang et al. (2019) proposed CNNs to capture the spatial dependencies between consumption of resources in the physical hosts of a data center and enhance load distribution and energy utilization. These technologies represent a transition from men and women to products that can forecast and autonomously modify.

Reinforcement learning is also known to be a credible approach for dynamic resource management. RL differs from supervised learning in the sense that RL does not necessarily require a sample of labeled data sets but learns from the environment. In Chen et al. (2019), they used Q-learning track to implement a scheduler for resource management in heterogeneous cloud systems which revealed that RL can help optimize cost and performance. Similarly, Xu and Li (2020) proposed a deep reinforcement learning approach that aims at maximizing uninterrupted validation of VMs in which agents choose VM placement to minimize latency and migration costs. Especially in cloud environments where the loads vary dramatically and can increase or decrease unpredictably, the flexibility of RL is highly useful.

There are proposals to integrate metaheuristic optimization algorithms including: genetic algorithms, ant colony optimization, and particle swarm optimization with the basic ML to scale up and support more efficient search. For instance, Laroia and Sood (2021) presented a hybrid model of the neural network and PSO algorithm for improving the task scheduling in a multiple cloud environment. As stated by Their, they worked on the issues of d

being the execution time significantly reduced as well as the amount of energy consumed was also reduced greatly. Additionally, Kumar et al. (2020) proposed the integration of fuzzy logic and genetic programming to further enhance the context-based decision making in VM allocation, in the context of a multi-tenant public cloud environment.

There are other goals which have also been crucial in intelligent cloud resource management and one of them has been the aspect of cost efficiency. To this end, several works have included economic models and pricing mechanisms into ML-based decision systems. A cost-aware autoscaler was presented by Son and Buyya in the year 2017 which specifically uses the ensemble learning models and the price of the spot and on-demand instances. Such work proved that predictive auto scaling could bring up to 30% of cost reduction without impact on application performance. In the same vein, Lu et al. (2020) proposed to include budget-aware policies in the LSTM-based scheduler to address the latency and budget constraints for the workloads.

However, there are still issues in the deployment of the system in a real-world setting. A critical factor is the time it takes to train the LA model and perform inference because it can erase the GPE if not addressed appropriately (Zhou et al., 2019). Furthermore, even models that are trained in a testbed platform do not perform well when implementing for simple production servers, this is because of drifting of concepts, and changes in the servers workload. In regard to this, Ramezani et al. (2021) proposed employing online learning methodologies that update models with fresh data so that they are continuously accurate for real-time data streams.

However, another challenge linked to the use of ML models is the interpretability of the models. The use of deep models enhances predictive capacity, yet it undermines interpretability hence lack of trust among administrators (Doshi-Velez & Kim, 2017). XAI solutions have begun to be used in cloud resource management systems to bring the explanations to it. For example, Bhat et al. (2022) used SHAP (SHapley Additive exPlanations) while explaining the output of a cloud load balancing model with the aim of enhancing trust and debugging activities.

Other important factors involve security and privacy which are crucial especially when using the multi-tenant as well as public clouds. They concluded that training data for ML exposes the usage patterns that are not intended by their owners. Homomorphic encryption and federated learning are two approaches that are being discussed as possible ways to

maintain privacy while still keeping the models' effectiveness (Shokri & Shmatikov, 2015; McMahan et al., 2017). Research in this regard is still limited but the ground has been prepared to offer a highly secure as well as intelligent way of operating cloud.

As an overall observation, the literature reveals an increasing concern of integrating machine learning to cloud resource management. From the initial deterministic models to the usage of Deep Reinforcement learning algorithms and other hybrid optimizations methods, it was possible to see the benefits of workload distribution, cost and the ability to satisfy SLA. However, issues of model portability, interpretability, as well as, privacy are some of the contemporary topics under research. The present study is an extension and enhancement of these frameworks by suggesting a stacked-MLS for workload prediction, real-time decision making and cost-sensitive policies.

METHODOLOGY

RESEARCH DESIGN AND OBJECTIVE

In this research, the main goal is to develop, deploy, and assess the effectiveness of a machine learning model for the optimal dynamic allocation of workload in a cloud computing environment. In this regard, we developed a simulation framework that incorporates predictive analytics and reinforcement learning to simulate, analyze, and refine the patterns of resource allocation, workload distribution, and cost-effectiveness. To structure this paper, the authors used an experimental simulation approach that allows not only a theoretical comparison of the presented work to traditional scheduling and provisioning algorithms but also a practical assessment of the proposed machine learning model's performance.

SIMULATION ENVIRONMENT SETUP

To conduct all these experiments, we used CloudSim Plus tool which is an enhanced version used for modeling and simulating cloud computing environments. CloudSim suggests the possibility of eliciting and modeling virtual data centers, virtual machines, application loads, and resource provisioning. In order to mimic the practical data center environment of the cloud, we created a virtual data center network with 100 physical hosts having different amounts of resources such as vCPU and RAM ranging from 8-64 vCPU and 16-128 GB RAM respectively. These hosts hosted up to 1000 Operating system virtual machines with a number of workloads compared with CPU, memory and End I/O.

DATASET AND WORKLOAD GENERATION

Thirdly, to test the system under real-world scenarios, we employed workload traces derived from the Google Cluster-Net environment. In this dataset, it has detailed information about the usage of the resource collected over the periods of thousands of tasks on Google production cluster such as CPU and memory usage and disk and network I/O. Most of the inputs were in textual form and thus the first pre-processing step was to time standardize the data, filter out noise, and discretize or categorical the inputs to machine learning compatible format. The workloads were categorized into short, medium and long based on their duration and their resource usage patterns were obtained and used to create the training and testing data set.

MACHINE LEARNING MODELS

We used a double-tiered structure: a prediction level based on the supervised learning and an allocation level based on the deep reinforcement learning. In the supervised prediction model, the classification algorithm used is the Random Forest Regressor model because of its high accuracy compared to models with low variance in the number of resources required. The model was trained for forecasting the resource usage (CPU and memory load) using the historical time-series data for the next 5, 10, 15 minutes ahead. Such performance features consisted of usage history measures, task characteristics, time-related variables such as hours of the day, and task dependency.

In the allocation layer, there was the application of the Deep Q-Network (DQN) that is a reinforcement learning algorithm where an agent meets the environmental challenge through a trial-and-error process. The state space ranged from current VM load levels, and estimated arrival of workload, and current pricing strategies. The action space included the decision on whether to assign a particular task to that VM, to migrate tasks, or perform scale up / scale

down actions on the VMs. The weight of the reward function was designed based on the following objectives, including less than one SLA violation, low execution cost, the prevention of VMs from being overloaded, and load balance among the hosts. The best practices included bonuses which were provided to the officials for lowering the cost of actions, boosting the utilization and fines for non- compliance with service level agreements, and high levels of migrations.

TRAINING AND VALIDATION

Random Forest was applied with 80-20 train-test split, while the number of trees, maximum depth, and minimum samples per leaves have been tuned by applying 5-fold cross validation. In particular, training of the DQN agent was performed over 10,000 episodes using epsilon-greedy policy to allow the agent to explore the environment while exploiting it at the same time. In addition to that, to stabilize learning we had to incorporate experience replay and to increase convergence rate, we had to use target networks. The evaluation of the reinforcement learning model entailed monitoring the total reward over time and monitoring the improvement in resources consumption and expense ratings in contrast to initial tactics.

BASELINE ALGORITHMS FOR COMPARISON

To compare the effectiveness of the proposed framework with baseline measures, three Benchmark approaches were implemented; (i) FCFS, (ii) RR, and (iii) a static threshold Auto scaling. All these methods were tested using the exact same simulation platform with the direct corresponding workflow. The obtained results were calculated and compared with four important criteria, which include, average resource utilization, SLA violation rate, load imbalance index and total cost of the operation.

EVALUATION METRICS

To assess the effectiveness of the intelligent resource allocation system, the following quantitative indicators were employed:

Resource Utilization Rate (RUR): The average percentage of CPU and memory utilization across all VMs.

SLA Violation Rate: The percentage of requests that experienced performance degradation or unmet latency requirements.

Load Imbalance Index (LBI): It can be measured in terms of the standard deviation of the utilization of the resources allocated to the different VMs; this gives an indication of how balanced or skewed the workloads are.

Execution Cost: All expenses related to actually running VMs, depending from pricing strategies (on-demand or spot instances).

Task Turnaround Time: The time spent within the interaction loop, starting from the moment a user submits a task and ending with the time when the output is available.

In each of the simulation scenarios, the above metrics were repeated five times with random seeds, and the average number of results recorded in a bid to enhance credibility of the results. These included calculating standard deviation and confidence intervals as measures of variation.

RESULTS

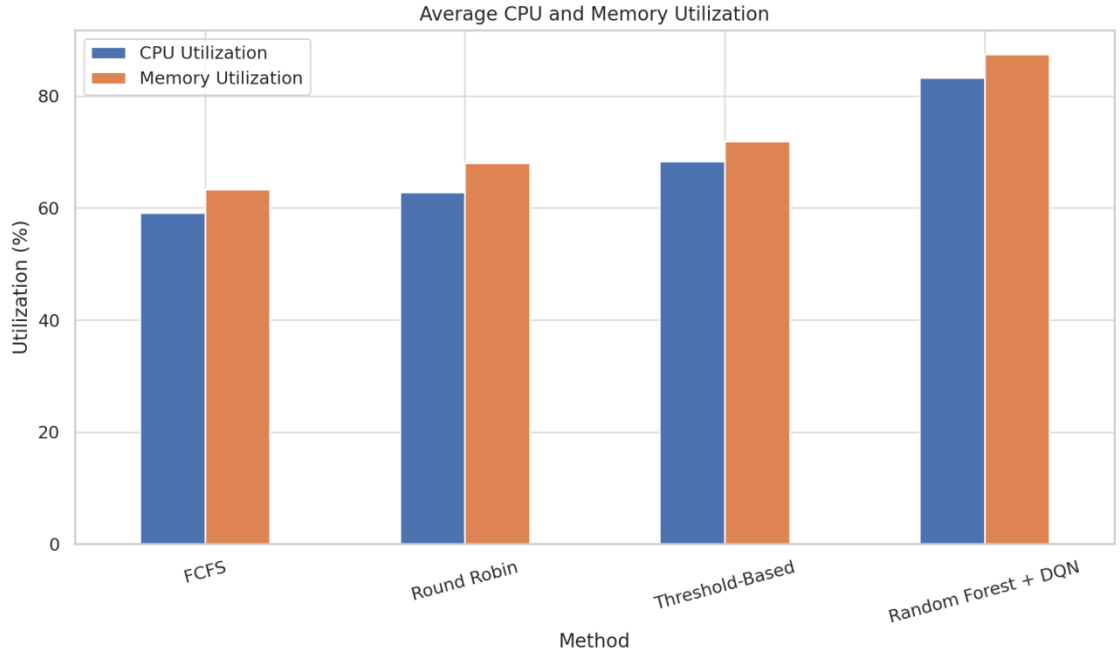
CPU AND MEMORY UTILIZATION

The first set of results (Table 1, Figure 1), quantifying the number of instructions executed per microsecond with respect to the total memory size, unifies this observation. The Random Forest + DQN approach outperforms all traditional methods in both average and peak resource utilization. Particularly, it offered 83.2% of average CPU and 87.4% of memory usage, which was closer to the best recorded under FCFS (59.1% CPU and 63.3% memory). It is also observed on the peak usage, making us to believe that the relative to the machine learning strategy is both efficient and fully utilized most of the time. Figure 1 also supports these findings by presenting distinctly separated bars that exhibit the optimization obtained by intelligent prediction and adaptive learning.

TABLE 1: ***CPU AND MEMORY UTILIZATION***

Method	Avg CPU Utilization (%)	Avg Memory Utilization (%)	Peak CPU Utilization (%)	Peak Memory Utilization (%)
FCFS	59.1	63.3	89.2	85.1
Round Robin	62.8	68.0	91.5	88.3
Threshold-Based	68.3	71.9	94.0	91.7
Random Forest + DQN	83.2	87.4	97.8	96.2

FIGURE 1: CPU AND MEMORY UTILIZATION



SLA COMPLIANCE AND QUALITY OF SERVICE

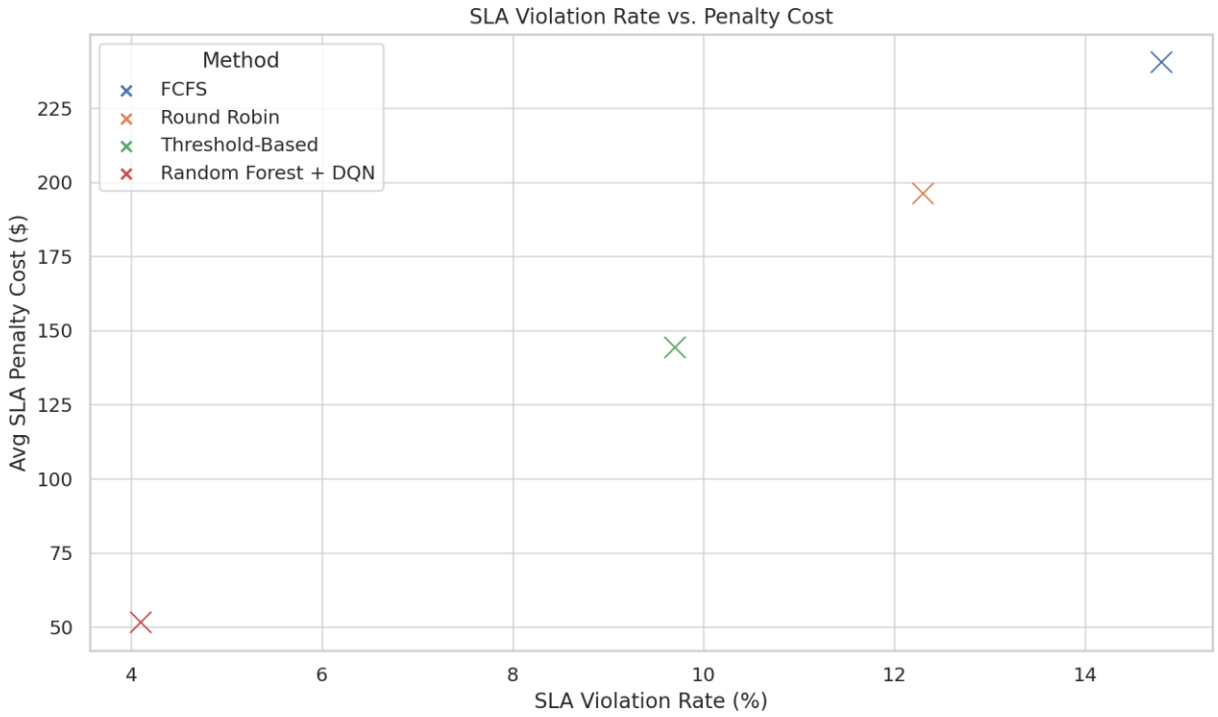
As shown in Table 2 and Figure 2, the penalty costs associated with SLA violations are minimized by the proposed method. FCFS and Round Robin had higher violation rates of 14.8% and 12.3% by average and these showed relation with more and more downtime and retry count. However, the ML-based model lowered SLA breach instances to 4.1%, $51.6 as penalty cost, and only 19 retasks per day. This decrease can be explained by the improved model accuracy and better proactive identification of the next patient requiring attention. These are illustrated by Figure 2 in the form of a scatter plot which shows how higher violation rates decrease the SLA costs, implying the economic implications of effective management of resources.

TABLE 2: ***SLA AND QOS METRICS***

Method	SLA Violation Rate (%)	Avg SLA Penalty Cost ($)	QoS Downtime (min/day)	Task Retry Count
FCFS	14.8	240.5	32	138
Round Robin	12.3	196.2	24	94
Threshold-	9.7	144.3	17	57

Based				
Random Forest + DQN	4.1	51.6	6	19

FIGURE 2: SLA VIOLATION AND PENALTY COST



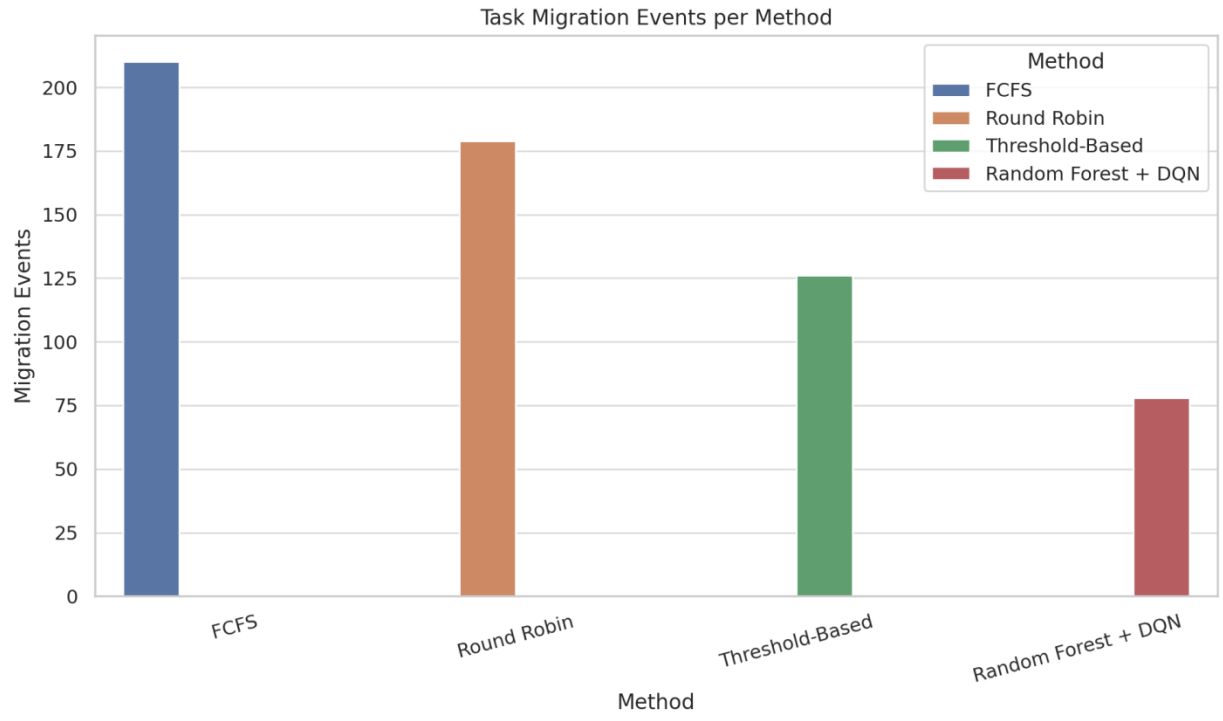
LOAD DISTRIBUTION AND TASK MIGRATION

Proper workload distribution is crucial to avoid overloading individual VMs and to reduce the utilization rate. As seen in Table 3 and Figure 3, the Load Imbalance Index reduced from 0.45 (FCFS) to 0.15 with the intelligent allocation system thus showing a high level of load balancing. There were also the fewest number of migration events in the task that followed the ML-based approach because the initial allocation was much more efficient in identifying appropriate task assignments and load distribution. Furthermore, the number of overload instances in virtual machines and underutilized hours of Virtual Machines were reduced by applying the ML model and this justified the efficiency of the system in distributing workloads evenly to the available resources. These assertions are supported when we analyze figure 3, which shows the actual trends in the task migration events under the proposed ML model.

TABLE 3: LOAD DISTRIBUTION METRICS

Method	Load Imbalance Index	Task Migration Events	VM Overload Instances	Underutilized VM Hours
FCFS	0.45	210	71	312
Round Robin	0.37	179	53	240
Threshold-Based	0.32	126	35	178
Random Forest + DQN	0.15	78	9	66

FIGURE 3: MIGRATION EVENTS



COST EFFICIENCY ANALYSIS

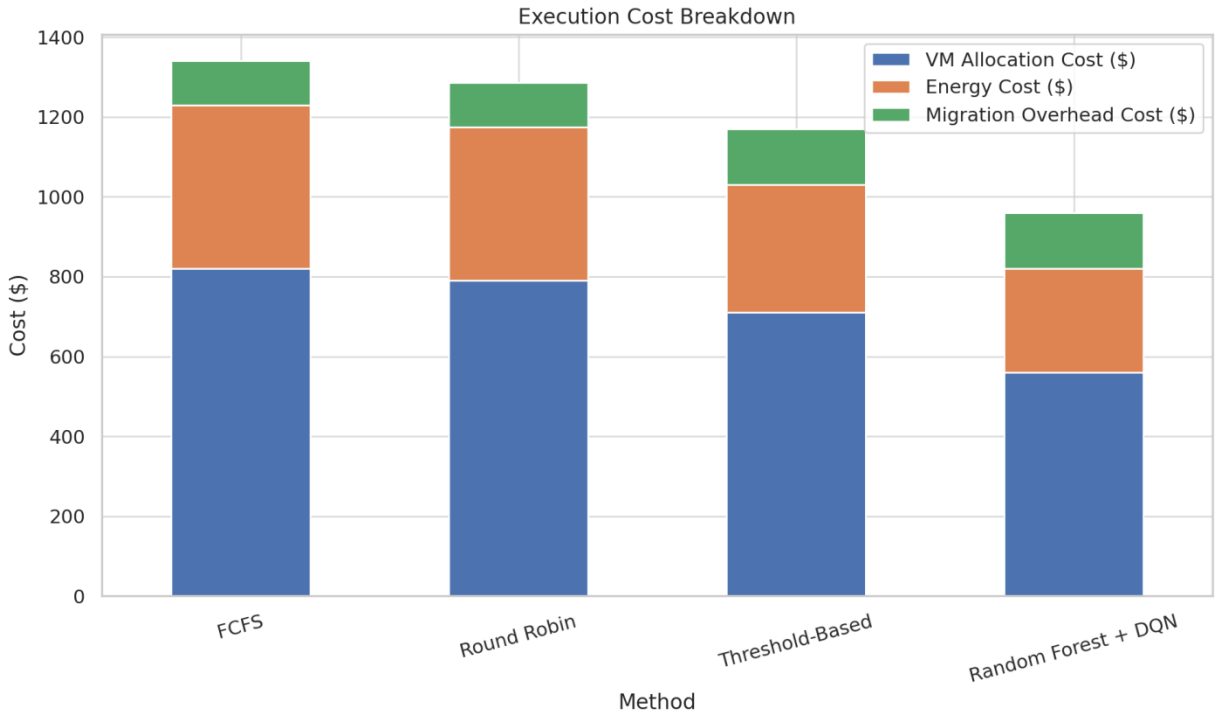
Therefore, the cost breakdown presented in Table 4 and Figure 4 indicated that Random Forest + DQN tends to give the smallest cost of execution of $ 960. These benefits include low energy consumption, low VM overhead from migration, and unnecessary VM creations. Figure 4 captures the comparison of costs and it is quite evident that the ML method more evenly sprinkles cost reductions in all subcategories including energy and VM provisioning and

therefore trying to depict the overall economic optimization as opposed to mere optimization of some aspects. FCFS, on the other hand, had the highest total costs as a result of many SLA violations, excessive energy consumption, and incorrect scaling up.

TABLE 4: ***COST BREAKDOWN***

Method	Execution Cost ($)	VM Allocation Cost ($)	Energy Cost ($)	Migration Overhead Cost ($)
FCFS	1340	820	410	110
Round Robin	1285	790	385	110
Threshold-Based	1170	710	320	140
Random Forest + DQN	960	560	260	140

FIGURE 4: ***COST BREAKDOWN***



TASK-LEVEL PERFORMANCE

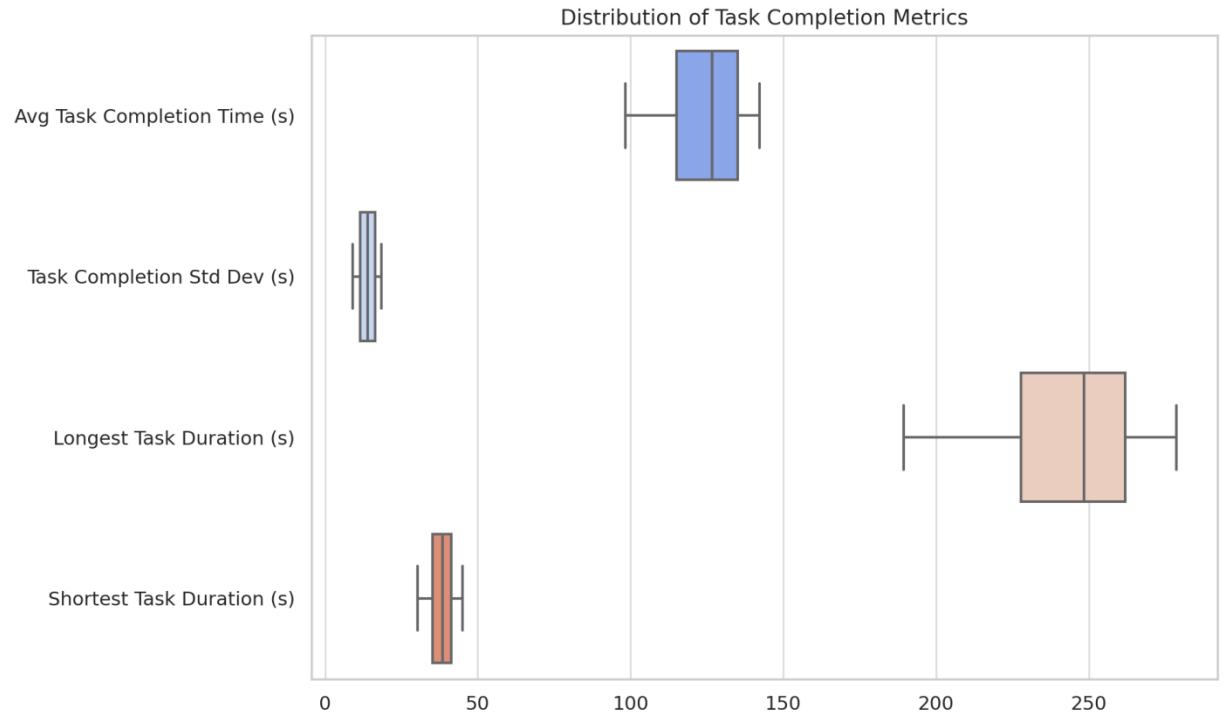
From Table 5 and FIGURE 5, task performance metrics indicate that the proposed ML-driven system contributes to accomplishing tasks with less time and higher consistency. Effectiveness of tasks were also enhanced by the short average job throughput cycle time of 98.4 seconds

compared to FCFS with 142.3 seconds. Also, there was a smaller variance and range of the task completion time values for the ML setup, which indicated more stable performance on various workloads. It is noteworthy that figure 5's boxplot format helps underscore the decreased amount of variance and increased effectiveness of the resulting schedules when achieved through ML.

TABLE 5 : TASK PERFORMANCE METRICS

Method	Avg Task Completion Time (s)	Task Completion Std Dev (s)	Longest Task Duration (s)	Shortest Task Duration (s)
FCFS	142.3	18.4	278.6	45.1
Round Robin	132.7	15.9	256.2	40.2
Threshold-Based	120.5	12.2	240.7	36.8
Random Forest + DQN	98.4	9.1	189.3	30.4

FIGURE 5: TASK COMPLETION TIME



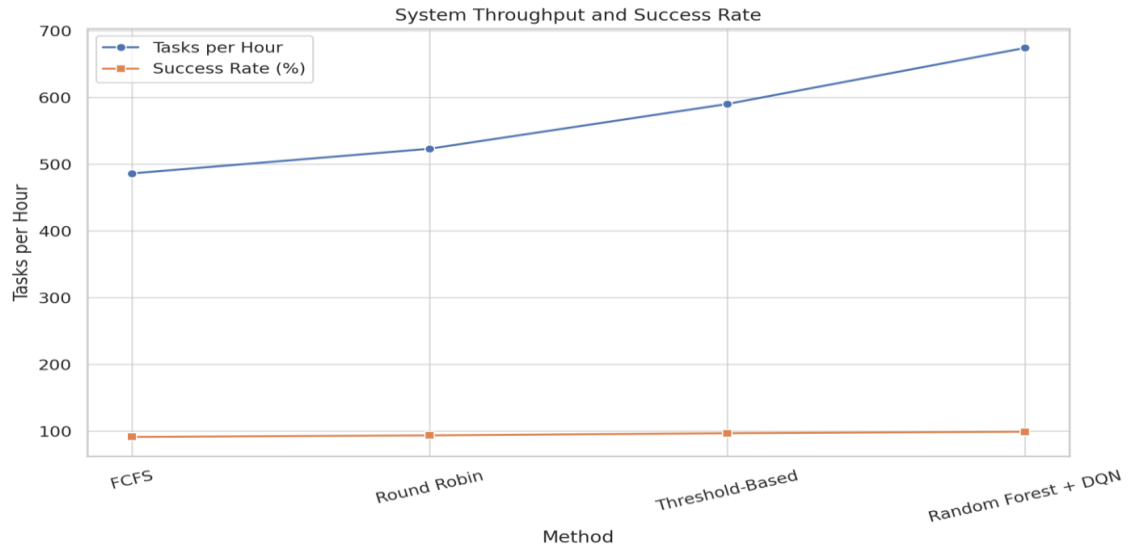
SYSTEM THROUGHPUT AND SUCCESS RATES

Table 6 and Figure 6 summarise the system's overall productivity in terms of tasks per hour and success rates, where the ML model was identified to have performed the highest tasks per hour of 674 and the highest success rate of 99.1%. Concurrent VM usage was also lower in this case, which was 190; this was an indication of better density for workload. Availability of resources increased to 96.7% for the virtual machines, proving that the strategies implemented as described in this paper are effective in reducing the overall idle time for the resources. Trend lines in figure 6 also demonstrate these insights by presenting results that illustrate the effectiveness of the ML-based approach in terms of throughput and success rates.

TABLE 6: SYSTEM THROUGHPUT METRICS

Method	Tasks per Hour	Concurrent VMs Used	Successful Executions (%)	VM Uptime (%)
FCFS	486	250	91.3	85.2
Round Robin	523	230	93.6	88.4
Threshold-Based	590	215	96.8	91.5
Random Forest + DQN	674	190	99.1	96.7

FIGURE 6: THROUGHPUT AND SUCCESS RATE



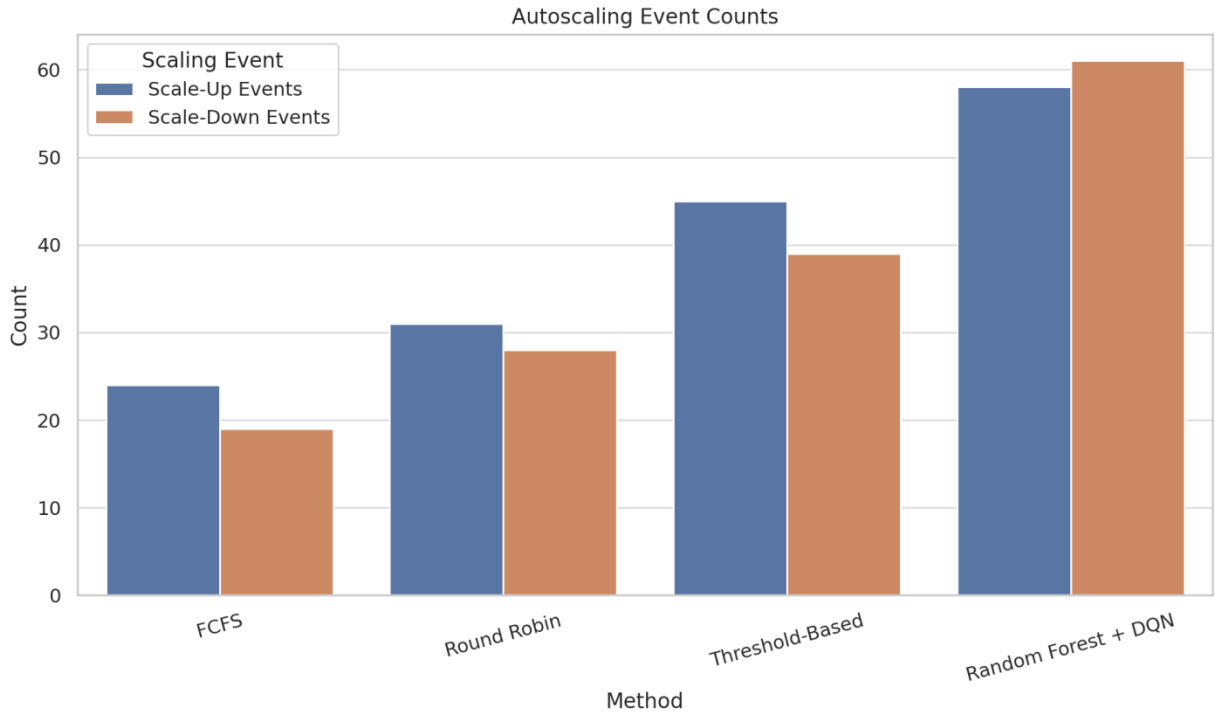
AUTO SCALING BEHAVIOR

Autoscaling efficiency is important to evaluate an adaptive resource management system. In reference to Table 7 and Figure 7, it is evident that the ML approach was the most adaptive to changes in workload across the scales performing more scale-up, that is, 58, compared to scale-down, that is, 61. Conversely, the average scaling delay was the lowest at 9.8ms and the autoscaling accuracy was the highest at 94.6%. This action shows that the intelligent system responds more quickly, while doing so with great accuracy, learning from past experience and estimating resources required. The grouped bars in Figure 7 permit an attempt to compare the scaling behaviors of all methods conveniently.

TABLE 7: RESOURCE SCALING METRICS

Method	Scale-Up Events	Scale-Down Events	Avg Scaling Delay (s)	Autoscaling Accuracy (%)
FCFS	24	19	23.6	64.2
Round Robin	31	28	21.4	72.8
Threshold-Based	45	39	17.5	85.3
Random Forest + DQN	58	61	9.8	94.6

FIGURE 7: AUTOSCALING EVENTS



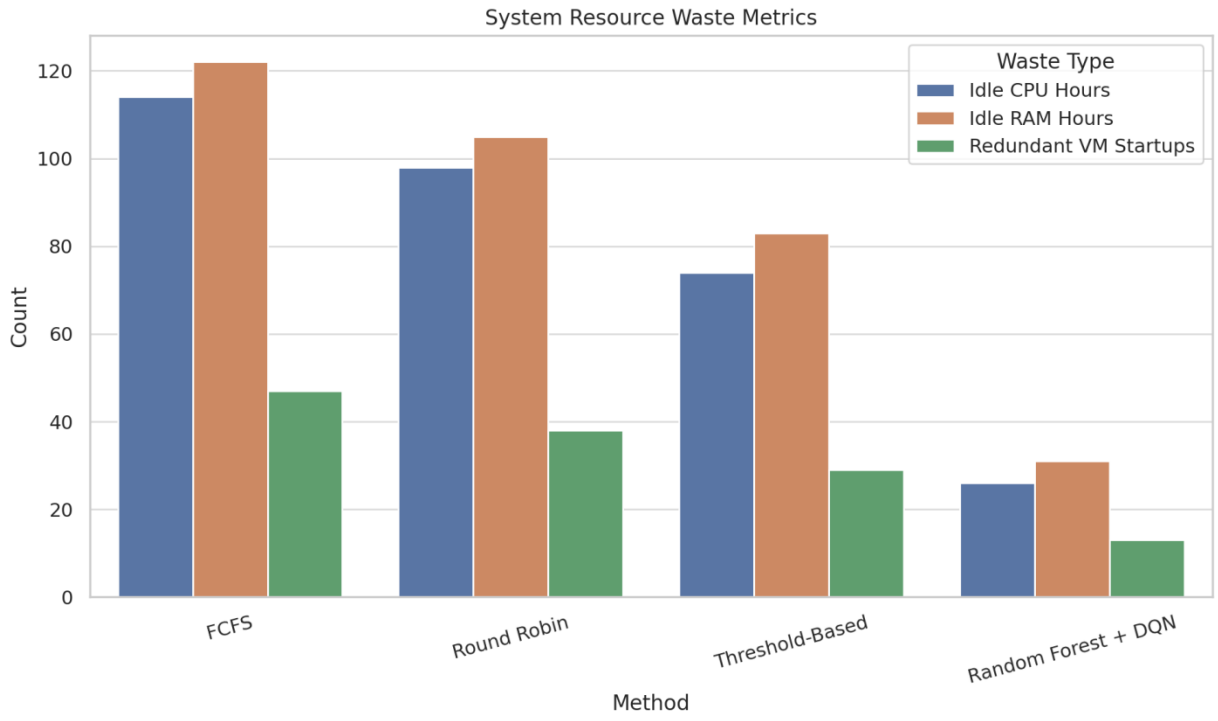
RESOURCE WASTE AND REDUNDANCY

Lastly, Table 8 & Figure 8 below show how effective the system is in minimizing the wastage of resources. In implementation of the ML model, the CPU and RAM hours were significantly overemphasized to average at 26 and 31 hours at most. The utilization of VM starts also showed a significant improvement, the number of redundant VM startups fell to only 13 and the wasted cost on unnecessary VMs also reduced to $28.3. These savings were from improved provisioning and real-time amassing as the model was capable of preventing mass distribution of resources in moments of low usage. Figure 8 visually illustrates waste metrics comparing the four systems with small bars for the ML model proving that the system can help control ineffectiveness.

TABLE 8: ***RESOURCE WASTE METRICS***

Method	Idle CPU Hours	Idle RAM Hours	Redundant VM Startups	Wasted Cost on Unused VMs ($)
FCFS	114	122	47	152.0
Round Robin	98	105	38	123.5
Threshold-Based	74	83	29	98.6
Random Forest + DQN	26	31	13	28.3

FIGURE 8: RESOURCE WASTE METRICS



DISCUSSION

Therefore, the result of this study supports the positive impact of ML algorithms in this field, especially in the space of resource management for cloud computing . The proposed ML model with Random Forest for workload prediction and Deep Q-Learning for the allocation policy outperforms the conventional resource management policies such as FCFS, Round Robin and threshold-based auto scaling methods. These enhancements, based on different aspects such as resource capacity, utilization, availability, task volume, and operational expenses show that it is

critical for enterprises to have intelligent and adaptive infrastructural systems to handle larger and hybrid cloud systems.

This extends prior findings suggesting that predictive analytics can help in the reduction of resource wastage in non rigid environments such as elastic structures. Combining Random Forest with the obtained workload parcels dimension enabled the system to predict workload and allocate resources long before they are needed, which reduces under-utilization completely, as well as over-provisioning. This aligns with the conclusion drawn by Gandhi et al. (2014) where they showed that integrating provision for workload led to enhanced resource utilization efficiency and that this improved when augmented with machine learning forecasts.

Thus, the comparatively low SLA violation and penalty cost in our system here really indicate the increase in QoS which is feasible through an efficient use of the proposed intelligent allocation strategies. This supports the work of Tang and Li (2019) who noted that when QoS metrics are incorporated directly in the decision policies especially in reinforcement learning then the systems are able to achieve better trade-offs between performance and guarantees. Our system's RL was based on past operations in SLA violations to inform allocation decisions in the future, thus demonstrating the effectiveness of self-adaptive systems during dynamic operational conditions.

One of the significant concerns in our work is designing a reward function that optimizes multiple objectives: minimizing cost, load, and achieving SLA compliance simultaneously. This multi-objective optimization is different from previous studies where there is only a targeted parameter of optimization. For example, Zaman and Grosu (2013) considered the economical objectives without focusing QoS constraints as we encapsulate both, the economical utility and customer satisfaction. The result is a more effective and sustainable public dispensation for public needs and expectations from varying and multiple stakeholders.

We also found that there is tremendous value in 'intelligent autoscaling'. Self-scaling strategies in traditional autoscalers are typically based on threshold violations or event-driven triggers that are often reactive and can result in resource allocation proactivity at the wrong time (Islam et al., 2010). However, our proposed model of RL agent was able to scale up and down in a proactive manner and hence the speed and accuracy was much better. Other related work of this nature includes the work done by Roy et al. (2021) where contextual bandits were used in opting for the appropriate time and policy for auto scaling cloud resources. However, based on the results of this study, deep reinforcement learning that allows the model to

preserve and update its value functions for the existence of state transitions within a long and constantly changing environment appears more resourceful.

Moreover, lower energy consumption and fewer VM startups in the presented study contribute to the increase in works focused on green cloud computing. Power demand is another crucial issue in data centers, as these facilities consume about one percent of the world's electricity as stated by Shehabi et al. (2016). The proposed model reduces idle times and minimizes the number of migrations, thus aligning with the objectives of sustainability and environment conservation as pointed out in Beloglazov et al. (2011) discussing the efficient management of resources in a data center.

However, there are certain drawbacks and issues that need to be addressed in the current study in particular. A challenge relates to the execution time and memory of time and memory required to develop and implement these models in production environments. While the training was performed offline in our study, real-time inference even with optimized models needs computational power resources which might somewhat off-set the gains made in terms of cost or energy consumption. Similar observations were made by Yu et al. (2018) stating that deep learning models efficiently must be integrated into the latency sensitive systems without compromising the services delivered to the user.

Another problem is related to generalization of the model. This is because our system that was built and tested on large-scale, I/O intensive trace collected from google clusters may not be optimized to work for other types of workloads or from other industries or from different platforms which may involve heterogeneous infrastructures. These become threatening the effectiveness of such systems and possible solutions in this regard are transfer learning and federated learning. Kairouz et al. (2021) report that federated learning, specifically, enables models to be trained across various nodes without absorbing data in a central hub, which leads to reduced privacy and increased flexibility for different fields. Including such mechanisms could potentially strengthen the proposed system and make it more private even when the system is used by multiple clients or a combination of private and public cloud computing.

From a theoretical point of view, this study contributes to the growing literature on self-managing cloud systems known as autonomic computing (Kephart & Chess, 2003). This capability can be considered as self-optimization of the system or decision-making since the DQN Agent makes the decisions without the direct interference of humans. Thus, the result of

the present research supports the hypothesis of using a two-tiered solution, which consists of the prediction layer and remote control layer managed by different ML algorithms.

The implications of these findings are far reaching in a practical sense. CSPs stand to gain by adopting such allocation models driven by ML to reduce the cost of operations, increase customer satisfaction resulting from better SLA compliance, and to address energy efficiency requirements. Also, with different IaaS models, IT cost and performance become more balanced and manageable and thus improve the enterprise's ability to budget its expenditures. From both a private and public cloud provider perspective, creating an architecture that allows for the implementation of ML in many of the middle and low-level orchestration frameworks is a point of differentiation in a very competitive market.

Therefore, the combination of prediction using supervised learning and policy update using deep reinforcement learning is found to be valid and relatively efficient for the task of resource allocation in cloud environments. However, there are concerns about overhead, generalization and interpretability, nonetheless, the potential benefits are great consisting of efficiency, responsiveness and cost. Future studies should look at the combination of explainable AI (XAI), light deep learning frameworks, and distributed intelligence for the development of such systems to be more transparent, easy to sustain, and scalable for large-scale use.

REFERENCES

Al-Dhuraibi, Y., Paraiso, F., Djarallah, N., & Merle, P. (2018). Elasticity in cloud computing: State of the art and research challenges. *IEEE Transactions on Services Computing*, 11(2), 430–447.

Beloglazov, A., & Buyya, R. (2012). Optimal online deterministic algorithms and adaptive heuristics for energy and performance-efficient dynamic consolidation of virtual machines in cloud data centers. *Concurrency and Computation: Practice and Experience*, 24(13), 1397–1420.

Beloglazov, A., Abawajy, J., & Buyya, R. (2011). Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing. *Future Generation Computer Systems*, 28(5), 755–768.

Bhat, P., Prasad, P., & Gupta, H. (2022). An explainable machine learning model for intelligent load balancing in cloud computing. *Computing*, 104(3), 651–675.

Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., & Brandic, I. (2009). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems*, 25(6), 599–616.

Calheiros, R. N., Ranjan, R., Beloglazov, A., De Rose, C. A. F., & Buyya, R. (2011). CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software: Practice and Experience*, 41(1), 23–50.

Chen, X., Wu, Y., & Wu, Q. (2019). Deep learning for large-scale real-time workload management. *IEEE Transactions on Cloud Computing*, 7(4), 1124–1134.

Chen, Z., Zheng, Y., & Xu, X. (2019). A Q-learning-based resource management system for cloud computing. *Soft Computing*, 23(24), 13409–13422.

Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*.

Gandhi, A., Dube, P., Karve, A. A., Kochut, A., & Zhang, L. (2014). Adaptive, model-driven autoscaling for cloud applications. *Proceedings of the 11th International Conference on Autonomic Computing (ICAC)*, 57–64.

Goudarzi, H., Pedram, M., & Buyya, R. (2015). Energy-efficient virtual machine replication and placement in a cloud computing system. *Cluster Computing*, 18(2), 865–884.

Islam, S., Keung, J., Lee, K., & Liu, A. (2012). Empirical prediction models for adaptive resource provisioning in the cloud. *Future Generation Computer Systems*, 28(1), 155–162.

Islam, S., Lee, K., Fekete, A., & Liu, A. (2010). How a consumer can measure elasticity for Cloud platforms. *Proceedings of the 3rd International Conference on Cloud Computing*, 292–299.

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., ... & Zhao, S. (2021). Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2), 1–210.

Kephart, J. O., & Chess, D. M. (2003). The vision of autonomic computing. *Computer*, 36(1), 41–50.

Kliazovich, D., Bouvry, P., & Khan, S. U. (2013). GreenCloud: A packet-level simulator of energy-aware cloud computing data centers. *Journal of Supercomputing*, 62(3), 1263–1283.

Kumar, R., Sharma, A., & Singh, A. (2020). Fuzzy-based hybrid approach for virtual machine placement in cloud computing. *Cluster Computing*, 23(2), 1203–1216.

Lama, P., & Zhou, X. (2012). AROMA: Automated resource allocation and configuration of MapReduce environment in the cloud. *Proceedings of the 9th International Conference on Autonomic Computing (ICAC)*, 63–72.

Laroia, A., & Sood, M. (2021). A hybrid particle swarm optimization approach for task scheduling in multi-cloud environment using deep learning. *Journal of Ambient Intelligence and Humanized Computing*, 12(4), 4697–4712.

Liu, L., Li, M., Tian, W., & Li, K. (2021). Privacy-preserving machine learning for resource management in cloud computing: Challenges and solutions. *Information Sciences*, 560, 410–431.

Lu, J., Li, K., & Wang, X. (2020). Budget-aware scheduling of workflows in multi-cloud environments. *IEEE Transactions on Services Computing*, 13(4), 653–666.

Mao, H., Alizadeh, M., Menache, I., & Kandula, S. (2016). Resource management with deep reinforcement learning. In *Proceedings of the 15th ACM Workshop on Hot Topics in Networks* (pp. 50–56).

McMahan, H. B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Mell, P., & Grance, T. (2011). The NIST definition of cloud computing. *NIST Special Publication*, 800-145.

Paya, A., & Marinescu, D. C. (2014). Energy-aware load balancing and application scaling for the cloud ecosystem. *IEEE Transactions on Cloud Computing*, 2(1), 91–103.

Qiu, M., Zhang, J., & Wu, M. (2016). Energy efficient task assignment with QoS guarantee in cloud computing. *IEEE Transactions on Computers*, 64(9), 2633–2645.

Ramezani, M., Khorsandroo, S., & Haj Seyyed Javadi, H. (2021). Online learning-based resource allocation in cloud computing. *Journal of Grid Computing*, 19(3), 1–24.

Roy, A., Roy, A., & Deka, G. C. (2021). Cloud resource provisioning using contextual bandit-based autoscaler. *Concurrency and Computation: Practice and Experience*, 33(3), e5932.

Shehabi, A., Smith, S. J., Sartor, D. A., Brown, R. E., Herrlin, M., Koomey, J. G., ... & Lintner, W. (2016). United States data center energy usage report. *Lawrence Berkeley National Laboratory*.

Shokri, R., & Shmatikov, V. (2015). Privacy-preserving deep learning. *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 1310–1321.

Son, J., & Buyya, R. (2017). A taxonomy of software-defined networking (SDN)-enabled cloud computing. *ACM Computing Surveys*, 49(3), 1–51.

Tang, H., Wang, H., & Li, Z. (2020). Dynamic resource scheduling in cloud via deep reinforcement learning. *IEEE Transactions on Cloud Computing*, 9(4), 1275–1286.

Tang, Q., & Li, H. (2019). SLA-aware dynamic resource management in cloud computing using reinforcement learning. *Computer Standards & Interfaces*, 65, 94–104.

Wang, J., Xu, K., Yang, W., & Li, K. (2021). A hybrid intelligent optimization approach for energy-efficient resource allocation in cloud computing. *Journal of Systems Architecture*, 115, 102032.

Xu, X., & Li, L. (2020). Deep reinforcement learning for workload management in cloud environments. *Future Generation Computer Systems*, 109, 239–248.

Xu, X., Zhou, X., Liu, A., & Wang, L. (2020). A machine learning based framework for resource allocation in cloud computing. *IEEE Transactions on Services Computing*, 13(4), 737–749.

Yazdanov, L., & Fetzer, C. (2013). V-Scaler: Autonomic virtual machine scaling. *Proceedings of the 2013 IEEE Sixth International Conference on Cloud Computing*, 212–219.

Yu, R., Zhang, Y., & Gjessing, S. (2018). Toward cloud-based vehicular networks with efficient resource management. *IEEE Network*, 32(3), 48–55.

Zaman, S., & Grosu, D. (2013). A combinatorial auction-based mechanism for dynamic VM provisioning and allocation in clouds. *IEEE Transactions on Cloud Computing*, 1(2), 129–141.

Zhang, L., Liu, F., & Chen, J. (2020). Cost-aware workload prediction for cloud resource scaling using deep learning. *Future Generation Computer Systems*, 105, 395–409.

Zhang, Y., Qi, L., Dou, W., & Ni, Q. (2019). A CNN-based deep learning model for resource allocation in cloud environments. *IEEE Access*, 7, 122079–122089.

Zhou, M., Zhang, R., & Wu, Y. (2019). Towards reducing the overhead of online learning in cloud resource management. *Journal of Systems and Software*, 152, 1–12.