Scientific Publications

Journal of Artificial Intelligence and Big Data, 2022, 2, 1344
www.scipublications.com/journal/index.php/jaibd
DOI: 10.31586/jaibd.2022.1344

Article

Towards the Efficient Management of Cloud Resource Allocation: A Framework Based on Machine Learning

Jaya Vardhani Mamidala [1,*], Sunil Jacob Enokkaren [2], Avinash Attipalli [3], Varun Bitkuri [4], Raghuvaran Kendyala [5], Jagan Kurma [6]

[1] Department of Computer Science, University of Central Missouri, USA
[2] ADP, Solution Architect, USA
[3] Department of Computer Science, University of Bridgeport, USA
[4] Software Engineer, Stratford University, USA
[5] Department of Computer Science, University of Illinois at Springfield, USA
[6] Computer Information Systems, Christian Brothers University, USA

*Correspondence: Jaya Vardhani Mamidala (mvardhini29@gmail.com)

Abstract: In the constantly evolving world of cloud computing, appropriate resource allocation is essential for both keeping costs down and ensuring an ongoing flow of apps and services. Because of its adaptability to specific tasks and human behavior, machine learning (ML) is a desirable choice for fulfilling those needs. This study Efficient cloud resource allocation is critical for optimizing performance and cost in cloud computing environments. In order to improve the precision of resource allocation, this study investigates the use of Long Short-Term Memory (LSTM). The LSTM model achieved 97% accuracy, 97.5% precision, 98% recall, and a 97.8% F1-score (F1-score: harmonic mean of precision and recall), according to experimental data. The confusion matrix demonstrates strong classification performance across several resource classes, while the accuracy and loss curves verify steady learning with minimal overfitting. The suggested LSTM model performs better than more conventional ML (machine learning) models like Gradient Boosting (GB) and Logistic Regression (LR), according to a comparative study. These findings underscore the LSTM (Long Short-Term Memory) model's robustness and suitability for dynamic cloud environments, enabling more accurate forecasting and efficient resource management.

Keywords: Cloud Computing, Resource Allocation, Machine Learning, Reinforcement Learning, Deep Q-Learning

How to cite this paper:
Mamidala, J. V., Enokkaren, S. J., Attipalli, A., Bitkuri, V., Kendyala, R., & Kurma, J. (2022). Towards the Efficient Management of Cloud Resource Allocation: A Framework Based on Machine Learning. *Journal of Artificial Intelligence and Big Data, 2*(1), 165–175.
DOI: 10.31586/jaibd.2022.1344

Received: September 12, 2022
Revised: October 30, 2022
Accepted: November 29, 2022
Published: December 27, 2022

1. Introduction

The ability of cloud-based information systems to store data makes them essential in today's digital world, process, and manage vast quantities of data. With the ability to adjust to evolving task requirements, these systems provide on-demand, scalable tools [1]. This renders them ideal for a diverse array of applications, including enterprise-level software and applications designed for individual users [2,3]. However, this freedom also means that you must be skilled at managing resources to ensure the system operates efficiently, is cost-effective, and maintains user satisfaction. In cloud environments, resource allocation is typically determined by simple formulae or heuristics, however these approaches might not be able to effectively handle fluctuating demand [4,5]. Resource allocation is one of the main issues with cloud computing, i.e., allocating virtualized computing resources (CPU, memory, storage, bandwidth) to contending tasks and users in an optimum manner [6,7].

In real-world setups, cloud environments are characterized by extremely dynamic and unpredictable workloads, where user demands may fluctuate rapidly over short time intervals. In such settings, traditional resource allocation techniques, which are usually rule-based or threshold-based policies, are not adequate [8]. These are not adaptive and are slow to respond to sudden fluctuations in demand, resulting in resource utilization, overprovisioning, latency in services, and increased operational expenses [9]. Machine learning (ML) has emerged as an effective solution to these challenges, as it enables the implementation of optimisation strategies, real-time decision-making, and predictive analytics. Additionally, the prediction of patterns and resource optimality can be facilitated by the use of ML algorithms, resulting in improved overall resource efficiency [10]. ML has introduced transformative approaches to resource management. Predictive and adaptive resource allocation is facilitated by ML, which enhances efficiency and performance by utilising historical data and advanced algorithms.

2.1. Motivation and Contribution

It is impossible to exaggerate the importance of cloud-based information systems in today's digital environment, as they help organizations to handle, store, and process large-scale data effectively. Despite its scalability and flexibility, another key issue remains in the optimal assignment of virtualized computing resources, such as CPU, memory, storage, and bandwidth, in highly dynamic and unpredictable workload patterns. Traditional resource allocation schemes, typically implemented through static assignment rules or threshold-based heuristics, often fail to respond dynamically to sudden demand changes, resulting in inefficient resource usage, overprovisioning, delayed responses, and other operational issues. This shortcoming explains why smarter, dynamic mechanisms should be sought. Driven by these issues, the use of ML has become eminent and provides data-driven methods for predictive analytics and real-time decision-making. By leveraging past usage patterns and utilizing ML-based approaches, dynamic and optimized resource administration is achieved, thereby enhancing system efficiency, cost-effectiveness, and service dependability in cloud computing systems. The research makes a significant contribution to the following aspects of the cloud environment:

- Employed a realistic cloud operations dataset with 19 features and 4,000 records, reflecting diverse resource allocation scenarios.
- Implemented effective pre-processing steps including missing value handling, noise removal, and data standardization to enhance prediction reliability.
- Capitalised on the capacity of the LSTM model to understand sequence dynamics and long-term dependencies, which are essential for predicting cloud resource demands.
- Designed the model to predict and allocate resources dynamically, improving efficiency in real-time cloud environments.
- Evaluated model accuracy using multiple metrics (F1-score, precision, recall, and accuracy) for thorough performance assessment.

2.2. Justification and Novelty

The legitimacy of employing the LSTM model is substantiated by its successful capture of temporal dependencies and temporal patterns, which are critical factors in the effective allocation of cloud resources in a constantly evolving environment. In contrast to traditional models like Logistic Regression (LR) and Gradient Boosting (GB), which do not account for dynamic relationships, LSTM may gain knowledge from historical trends and patterns of usage growth and decline. The use of a deep learning (DL) based sequential model to forecast cloud resources, which is more precise and broadly applicable, is what makes this study innovative. This approach enables more adaptive and intelligent allocation, reducing both over-provisioning and under-provisioning of resources, which is critical for optimizing performance and cost in cloud systems.

2.3. Paper Organization

The paper is organized as: Section II presents the literature study in resource allocation using ML. Section III present the research methodology in detail. The experiment results and comparative analysis are present in Section IV. Conclusion with limitations and future work in section V.

3. Literature Review

A wide range of significant research studies on Efficient Cloud Resource Allocation have been reviewed and analysed to guide and support the development of this work.

Chudasama and Bhavsar (2020) highlights the importance of resource elasticity in cloud applications. Traditional adaptive policies, such as threshold-based auto-scaling, may not be effective during dynamic workloads. They provide a method to forecast short-term computing resource consumption based on queuing theory and DL. The proposed model improves resource elasticity and performance metrics, outperforming the baseline model by 5% [11].

Chen et al. (2019) The distribution of cloud-based software service resources is proposed to be self-adaptive and self-learning. By using machine learning to build a QoS model using historical data, the approach predicts the QoS value depending on the workload and resources allotted. The model will automatically make decisions on resource allocation through a genetic algorithm. The method has been tested on the RUBiS benchmark, achieving an accuracy of above 90 percent and a range of 10 to 30 percent improvement in resource utilization [12].

Rayan and Nah (2018) propose the development of machine learning-based methodologies that can predict the daily workload of operations in cloud data centres. Three methodologies are investigated: RFR, SVR, and polynomial regression. According to the research findings, RFR performs best, with a PC prediction of 4869.08 and a minimal root-mean-square error of 11.68. This assists in the management of resources, conservation of energy, saving of CPU and memory and better service. [13].

Ataie et al. (2017) suggest a compromise model to process large data sets using commodity hardware clusters, a system that would merge MapReduce and Apache Hadoop. In order to optimise precision at a minimal expense, this approach implements support vector regression and queuing networks. The experimental results suggest that the accuracy is increased by 21% in comparison to machine learning methods that do not employ analytical models [14].

Dai et al. (2016) propose a cloud computing multi-objective optimisation technique that aims to balance cost, availability, and performance for large-scale data applications. The method, following the analysis and modeling of the objectives involved, is 20% faster than conventional approaches, 15% more efficient in performance than other heuristic algorithms, and achieves a cost savings of 4-20% [15].

A summary of recent studies on the efficient management of cloud resource allocation using the ML approach can be found in Table 1, which presents innovative models, datasets used, and the main findings and challenges.

Table 1. Overview of Recent Studies on Cloud Resource Allocation Using Machine Learning

Author	Proposed Work	Dataset	Key Findings	Challenges/recommendation
Chudasama and Bhavsar (2020)	DL + Queuing Theory model for proactive auto-scaling	University server logs	Improved SLA violation prediction by 5%, Enhances resource elasticity under hybrid cloud	Static threshold auto-scaling fails under unpredictable loads, need for proactive, prediction-driven auto-scaling mechanisms in hybrid cloud environments
Chen et al. (2019)	A self-adaptive system for allocating resources for cloud-based software applications and self-learning, utilizing genetic algorithms for optimization and machine learning for QoS modelling.	RUBiS benchmark	QoS prediction accuracy > 90% 10%–30% improvement in resource utilization	Traditional policy-driven methods lead to complexity and high administrative cost; recommends ML-driven automatic decision-making to adapt to dynamic environments.
Rayan and Nah (2018)	ML-based workload prediction for cloud data centers (RFR, SVR, PR)	Operational workload logs	RFR achieved lowest RMSE (11.68 for PMs, 4869.08 for PC), 2-second training time Enables proactive allocation and energy/resource efficiency	Focused on prediction, not dynamic real-time scheduling, Need to integrate accurate workload prediction with adaptive scheduling/auto-scaling mechanisms in large-scale environments
Ataie et al. (2017)	Hybrid methodology that integrates support vector regression (SVR) and queuing networks to forecast the duration of job execution	Hadoop MapReduce job traces	Achieved 21% improvement in prediction accuracy over standalone ML methods	Need to balance accuracy and computational cost, Integration of analytical models and ML recommended for better resource management
Dai et al. (2016)	A method for multi-objective optimization that is intended to maximize the price, accessibility, and efficiency of cloud-based Big Data programs. carried out on the testbed.	Experimental setup	Execution time improved by 20% over traditional methods- 15% higher performance than heuristics- 4–20% cost savings	Emphasizes the need for fine-grained resource allocation in cloud infrastructure; recommends multi-objective optimization to handle competing objectives.

3. Research Methodology

The proposed cloud resource allocation model methodology is initiated by the data collection process, which involves sampling 19 columns and 4000 rows of data that represent the complexity of operations in the cloud. This is preceded by data pre-processing, which entails treating missing values by either deleting or imputing them, and eliminating noise to remove irrelevant or redundant data. Additionally, data standardization is performed to prevent interference with data prediction accuracy. The model's performance is further tested by dividing the dataset 80:20 across training and test sets. The Long Short-Term Memory (LSTM) model, which offers a significant advantage in handling temporal correlations and sequential patterns to identify optimal and adaptive cloud resource allocation, is finally put into practice. Finally, the accuracy, precision, recall, and F1-score are used to evaluate the model's performance in the context of ML-based cloud resource capacity allocation. Figure 1 displays the flowchart of phases for the resource allocation methodology.

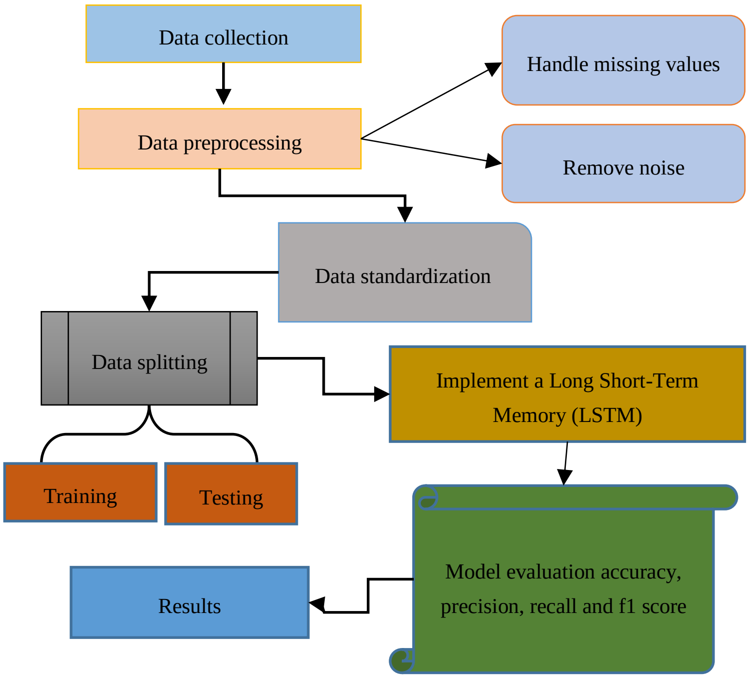

Figure 1. Proposed flowchart for Cloud Resource Allocation

The following steps involved in the proposed flowchart of detecting the efficient allocation of cloud resources will be described and discussed below.

3.1. Data Collection

The process of gathering full information on historical data from cloud service providers. This comprises operating expenditures, workload profiles, activity levels, and resource utilisation performance data. The dataset is used as the input source for training the suggested methods and has 19 columns and infinite attributes. This Figure reflects the intricacy of the resource allocation mechanism and is closely correlated with the dataset's row count, reaching 4000.

3.2. Data Pre-Processing

The data is collected, and then there is a systematic pre-processing stage of the data. It includes data cleaning to remove missing, noisy and inconsistent data, and data transformation to transform raw data into meaningful features that can be used in ML. To enable accurate model evaluation and ensure that the outcomes can be effectively used in scenarios that are not visible, the cleaned data is then separated into testing, validation, and training categories. The pre-processing steps that follow are the following ones:

- **Handle missing value:** The methods for handling missing data include imputation, which substitutes statistical estimators like the model, mean, or median of the missing data; and deletion, which removes rows with missing values. This is a crucial stage in determining the calibre of data used to train ML models.
- **Remove noise:** Data pre-processing uses the elimination of noise whereby irrelevant or unnecessary data points are detected and deleted as they do not add to the general structure of the data set. In order to mitigate noisiness in cloud

resource allocation, there are several strategies that you may utilise. The nature of noise, the choice of a proper model of cloud service, and the usage of dedicated or isolated resources can reduce the effects of noise.

3.3. Data Standardization

The data values are normalised by the utmost value in the same data feature, ensuring that they lie within the standard range of 0 to 1. The normalisation operation was implemented by employing Equation. (1). This process resulted in the conversion of all numerical values to a value range of 0 to 1.

$$X' = \frac{x-\mu}{\sigma} \tag{1}$$

The original feature value is denoted by x, while the normalised value is represented by X' in this equation. The standard deviation and mean are denoted by σ and μ, respectively. The normalisation process can mitigate the detrimental impact of features with high numerical values that would otherwise adversely affect performance.

3.4. Data Splitting

Data division is the process of dividing a dataset into smaller groups for 20% testing, 80% training, and 20% validation. Training comprises eighty percent of the allocation, while twenty percent is reserved for assessment.

3.5. Long Short-Term Memory (LSTM) Model

The classifier is capable of learning long-term dependencies between the text, which is why LSTM is particularly popular for text classification. LSTM classifiers are a type of recurrent neural network (RNN), which is a stratified network that employs the outputs of the preceding layer as inputs for the subsequent layer. There are feedback connections in LSTM that enable it to operate with sequences of data rather than merely individual data points [16,17]. An LSTM node is composed of a cell, input gate, output gate, and forget gate. Three gates control how information moves through the cell, which is in charge of holding onto values across time. Each memory block in the LSTM layers contains three multiplicative gates and is connected recurrently. To ensure that temporary data is used for a predetermined period, gates continuously write, read, and reset. The input of the unit, x_t, h_{t-1}, c_{t-1} and the output of the unit, h_t, c_tWe updated as follows Equation from (2) to (7):

Gates:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{2}$$

$$f_t = \sigma\left(W_f x_t + U_i h_{t-1} + b_f\right) \tag{3}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{4}$$

Input transform:

$$g_t = tanh\left(W_g x_t + U_g h_{t-1} + b_g\right) \tag{5}$$

State update

$$\mathrm{c_t = f_t \odot c_{t-1} + i_t \odot g_t} \tag{6}$$

$$\mathrm{h_t = o_t \odot tanh(c_t)} \tag{7}$$

In the previous equations, element-wise multiplication and the logistic sigmoid function are represented by σ and $\odot$, respectively. An input gate i_t, A forget gate f_t, an output gate o_t, a hidden unit h_t , and a memory cell c_t are present in the LSTM unit at each time step t. The learnt parameters are W and U, and the added bias is denoted by (b).

The input gate regulates how much each unit is updated, the forget gate regulates how much the memory cell is expunged, and the output gate regulates how much of the internal memory state is disclosed.

3.6. Evaluation Metrics

To compare the results by evaluating their accuracy and F1-scores after the pre-processing and modelling phases. Utilise a confusion matrix to ascertain these Figures. A confusion matrix is employed to ascertain F1-score, recall, accuracy, and precision. These metric values are derived from the training subset [18]. The highest True Negative (TN) and True Positive (TP) values are preferred. True Negative is a term that denotes situations in which the actual and anticipated data are both negative (0). True Positive denotes that the actual and anticipated data are both true positives (1). The equations below give the following of the matrix equation.

Accuracy: The number of accurately predicted samples divided by the total number of samples in a dataset is the definition of this statistic, it is given as Equation (8):

$$Accuracy = \frac{TP+TN}{TP+Fp+TN+FN} \tag{8}$$

Precision: Precision is a measure that determines the precision with which a given model generates optimistic forecasts. The statistic measures the proportion of positively recognised instances relative to the overall count of instances that were anticipated to test positive, it is expressed as Equation (9):

$$Precision = \frac{TP}{TP+FP} \tag{9}$$

Recall: This metric, also known as TPR or sensitivity, is a metric that quantifies the accuracy of the model in classifying positive samples from all possible positive samples. Mathematically, it may be expressed as Equation (10):

$$Recall = \frac{TP}{TP+FN} \tag{10}$$

F1 score: F1 scores combine accuracy and recall into one statistic, making them a suitable way to evaluate the model's performance. Mathematically, it is given as Equation (11):

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{11}$$

In conclusion, the model's accuracy and its general propensity to accurately predict the objective variable are evaluated by all of these measures.

4. Results and Discussion

This section presents the experimental findings and the resource allocation simulation environment. The hardware of the experimental platform is configured with an Intel Core i7-6500U CPU, 8 GB RAM, and 1 TB storage. Table 2 displays the experimental outcomes of the suggested LSTM model for cloud resource allocation. The model is highly efficient in terms of all essential performance indicators, achieving a 97 percent accuracy rate, 97.5 percent precision rate, 98 percent recall rate, and a 97.8 percent F1-score. These results demonstrate that the LSTM model can accurately anticipate resource consumption with a high recall level and little FP and FN occurrences. This demonstrates the model's effectiveness and appropriateness in addressing the dynamic and complex nature of cloud computing situations.

Table 2. Experiment Results of Proposed Models for Cloud Resource Allocation

Performance matrix	Long Short-Term Memory (LSTM)
Accuracy	97
Precision	97.5
Recall	98
F1-score	97.8

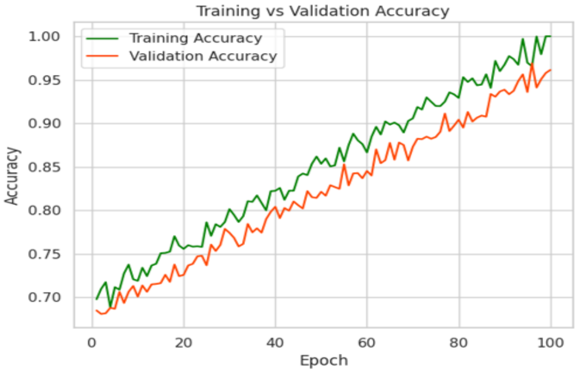

Figure 2. Accuracy Curves for the LSTM Model

As shown in Figure 2, the LSTM model with 100 training epochs has a training and validation accuracy diagram. The two curves exhibit a consistent upward trend, suggesting that the learners were progressively learning and improving over time. Training accuracy is generally higher than validation accuracy, although the difference is minor, as expected. This is a sign that the model is not overfitting to a significant extent. The training curve is tracked by the validation accuracy, which tends to converge to it in the concluding phases of the training. This nearly flawless outcome demonstrates how well the LSTM model classifies and generalises to previously unobserved data.

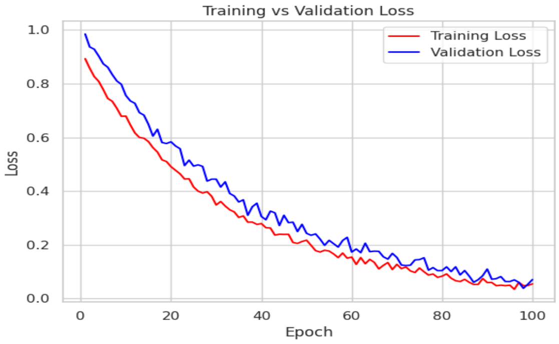

Figure 3. Loss Curves for the LSTM Model

Figure 3 shows the LSTM model's training and validation loss across 100 epochs. Learning has occurred, and the model is converging, as evidenced by the decreasing trend of both contours. The training and validation losses, which are initially substantial but rapidly decrease throughout the first epochs, show the model's ability to identify significant patterns in the data. Training progresses, and the loss reduction becomes more gradual, ultimately stabilising at or near zero. The validation loss is marginally greater than the training loss, which implies that the generalisation is adequate and that there is minimal overfitting. The LSTM model's robust predictive performance and sustained learning are both confirmed by the close alignment of both curves.

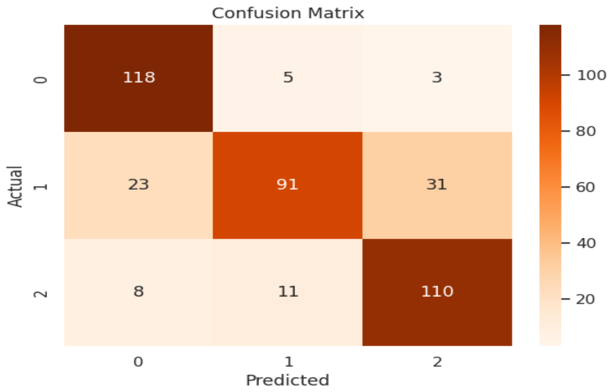

Figure 4. Confusion Matrix for LSTM Model

The efficacy of a multi-class classification model across three classes (0, 1, and 2) is illustrated in Figure 4, which is a confusion matrix. The diagonal elements (118, 91, and 110) represent the cases for each class that were accurately predicted, indicating excellent overall performance. However, some misclassifications are present: Classifying 23 instances of class 1 as class 0 and 31 instances of class 1 as class 2 was incorrect. Eleven Class 2 cases were mistakenly classed as Class 1, while eight Class 2 cases were mistakenly classified as Class 0. Because just five and three instances were misclassified, Class 0 had very little ambiguity. While class 1 has the most classification errors, indicating possible areas for development in differentiating this class, the model performs best in recognising class 0 and class 2.

Table 3. Comparison of different Cloud Resource Allocation using Machine Learning.

Models	Accuracy
Gradient Boosting (GB) [19]	92
Logistic Regression (LR) [20]	95
Proposed LSTM Model	97

In Table 3, the effectiveness of three different ML models—GB, LR, and LSTM—in relation to cloud resource allocation is compared, with accuracy serving as the assessment criterion. With respective accuracy rates of 92% and 95%, the GB and LR models demonstrated their capacity to handle structured data and perform rather well in resource allocation prediction. Nevertheless, although these models facilitate accurate capture of linear relationships and boosting-based enhancements, they might be inadequate to account complex temporal patterns reported in dynamic cloud spaces. Conversely, the LSTM model performed better, achieving 97% accuracy, which expresses its superior ability to learn and interpret long-term dependencies and nonlinear relationships in chronological data. This makes LSTM especially suitable for forecasting resources whose use varies over time and is dependent on specific needs.

The suggested LSTM model offers substantial benefits to cloud resource allocation prospects due to its ability to learn and model temporal patterns effectively in the usage dataset. LSTM is better at handling longer-term dependencies than traditional models, and this is essential in dynamic and time-based clouds. This will result in increased accuracy in the predictions, as shown in the experiment findings and more accurate forecasting of the resources requirements. Consequently, it aids in reducing both over-provisioning and under-provisioning, thus enabling the optimum use of resources.

5. Conclusion and Future Study

Reinforcement learning-based techniques have entered the industry as a result of cloud systems' requirement for dynamic resource allocation. The cloud computing services offer the users on demand resources of diverse workloads that demand different performance of services. Nevertheless, changing workloads, resource needs, and the conflict between effective performance and cost effectiveness may exert a severe burden on resource management in such kind of platforms. This investigation reveals that the accuracy and efficiency of allocating cloud resources using ML and DL models show tremendous potential. In the tested models, LSTM had displayed the highest accuracy (97%), compared to GB and LR which were 95% respectively. Although the suggested LSTM model for cloud resource allocation has shown promising results, several limitations apply. The model's effectiveness may be impacted by the amount and quality of the data collection, as in this study, no more than 4,000 records were employed. Future research will involve the use of larger real-time datasets and experiments on hybrid models, such as LSTM-Transformer, to achieve improvements in accuracy and scalability within dynamic cloud settings.

References

[1] P. Peddi and D. S. Arumugam, “Comparative study on cloud optimized resource and prediction using machine learning algorithm,” *Anveshana's Int. J. Res. Eng. Appl. Sci.*, vol. 1, no. 3, 2016.

[2] F. Nzanywayingoma and Y. Yang, “Efficient resource management techniques in cloud computing environment: a review and discussion,” *Int. J. Comput. Appl.*, 2019, doi: 10.1080/1206212X.2017.1416558.

[3] C. Riedelsheimer and A. E. Melchinger, “Optimizing the allocation of resources for genomic selection in one breeding cycle,” *Theor. Appl. Genet.*, 2013, doi: 10.1007/s00122-013-2175-9.

[4] M. Abbasi, M. Yaghoobikia, M. Rafiee, A. Jolfaei, and M. R. Khosravi, “Efficient resource management and workload allocation in fog–cloud computing paradigm in IoT using learning classifier systems,” *Comput. Commun.*, vol. 153, pp. 217–228, Mar. 2020, doi: 10.1016/j.comcom.2020.02.017.

[5] M. Aibin, “LSTM for Cloud Data Centers Resource Allocation in Software-Defined Optical Networks,” in *2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, 2020, pp. 162–167. doi: 10.1109/UEMCON51285.2020.9298133.

[6] A. Beloglazov, J. Abawajy, and R. Buyya, “Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing,” *Futur. Gener. Comput. Syst.*, 2012, doi: 10.1016/j.future.2011.04.017.

[7] M. Zamzam, T. Elshabrawy, and M. Ashour, “Resource Management using Machine Learning in Mobile Edge Computing: A Survey,” in *Proceedings - 2019 IEEE 9th International Conference on Intelligent Computing and Information Systems, ICICIS 2019*, 2019. doi: 10.1109/ICICIS46948.2019.9014733.

[8] N. Liu *et al.*, “A Hierarchical Framework of Cloud Resource Allocation and Power Management Using Deep Reinforcement Learning,” in *Proceedings - International Conference on Distributed Computing Systems*, 2017. doi: 10.1109/ICDCS.2017.123.

[9] A. Yousafzai *et al.*, “Cloud resource allocation schemes: review, taxonomy, and opportunities,” *Knowl. Inf. Syst.*, 2017, doi: 10.1007/s10115-016-0951-y.

[10] F. D. la Prieta, S. Rodríguez-González, P. Chamoso, Y. Demazeau, and J. M. Corchado, “An Intelligent Approach to Allocating Resources within an Agent-Based Cloud Computing Platform,” *Appl. Sci.*, vol. 10, no. 12, p. 4361, Jun. 2020, doi: 10.3390/app10124361.

[11] V. Chudasama and M. Bhavsar, “A dynamic prediction for elastic resource allocation in hybrid cloud environment,” *Scalable Comput.*, vol. 21, no. 4, pp. 661–672, 2020, doi: 10.12694:/scpe.v21i4.1805.

[12] X. Chen, J. Lin, B. Lin, T. Xiang, Y. Zhang, and G. Huang, “Self-learning and self-adaptive resource allocation for cloud-based software services,” *Concurr. Comput. Pract. Exp.*, vol. 31, no. 23, Dec. 2019, doi: 10.1002/cpe.4463.

[13] A. Rayan and Y. Nah, “Resource prediction for big data processing in a cloud data center: A machine learning approach,” *IEIE Trans. Smart Process. Comput.*, vol. 7, no. 6, pp. 478–488, 2018, doi: 10.5573/IEIESPC.2018.7.6.478.

[14] E. Ataie, E. Gianniti, D. Ardagna, and A. Movaghar, “A combined analytical modeling machine learning approach for performance prediction of MapReduce jobs in cloud environment,” *Proc. - 18th Int. Symp. Symb. Numer. Algorithms Sci. Comput. SYNASC 2016*, pp. 431–439, 2017, doi: 10.1109/SYNASC.2016.072.

[15] W. Dai, L. Qiu, A. Wu, and M. Qiu, “Cloud Infrastructure Resource Allocation for Big Data Applications,” *IEEE Trans. Big Data*, vol. 4, no. 3, pp. 313–324, Sep. 2018, doi: 10.1109/TBDATA.2016.2597149.

[16] Y. Liu, L. L. Njilla, J. Wang, and H. Song, “An LSTM Enabled Dynamic Stackelberg Game Theoretic Method for Resource Allocation in the Cloud,” in *2019 International Conference on Computing, Networking and Communications, ICNC 2019*, 2019. doi: 10.1109/ICCNC.2019.8685670.

[17] G. Park and M. Song, "Prediction-based resource allocation using LSTM and minimum cost and maximum flow algorithm," in *Proceedings - 2019 International Conference on Process Mining, ICPM 2019*, 2019. doi: 10.1109/ICPM.2019.00027.

[18] J. B. Wang *et al.*, "A Machine Learning Framework for Resource Allocation Assisted by Cloud Computing," *IEEE Netw.*, vol. 32, no. 2, pp. 144–151, 2018, doi: 10.1109/MNET.2018.1700293.

[19] S. D. Pasham, "Dynamic Resource Provisioning in Cloud Environments Using Predictive Analytics," vol. 4, no. 2, pp. 1–28, 2018.

[20] J. Zhang, N. Xie, X. Zhang, K. Yue, W. Li, and D. Kumar, "Machine learning based resource allocation of cloud computing in auction," *Comput. Mater. Contin.*, 2018, doi: 10.3970/cmc.2018.03728.