

Smart Cloud Architectures: The Combination of Machine Learning and Cloud Computing [†]

Aqsa Asghar ^{1,*}, Attique Ur Rehman ¹, Rizwan Ayaz ² and Anang Suryana ³

¹ Department of Software Engineering, University of Sialkot, Sialkot 51040, Pakistan; attique.urrehman@uskt.edu.pk

² School of Computer Science, Taylor's University, Subang Jaya 47600, Malaysia; rizayazuddin@gmail.com

³ Department of Electrical Engineering, Nusa Putra University, Sukabumi 43152, West Java, Indonesia; anang.suryana@nusaputra.ac.id

* Correspondence: aqsaasghar522@gmail.com

[†] Presented at the 7th International Global Conference Series on ICT Integration in Technical Education & Smart Society, Aizuwakamatsu City, Japan, 20–26 January 2025.

Abstract

Machine learning (ML) in cloud architectures is used to manage powerful servers that run distributed systems over the internet. ML predicts the workload and traffic from cloud consumers and allocates resources according to the demand. ML in cloud architectures is there to improve performance and increase availability to manage cloud computing resources. The combination of ML and cloud architectures balances the workload and ensures reliability. This research discusses cloud architectures that use ML to run different algorithms to predict the improvement in the cloud architectures by using a cloud computing resource dataset. The dataset is used with different classifiers with the same ML framework that is discussed in this paper; the ML framework has a sequence to provide the steps of the model training and testing and uses different techniques and methods for the better performance of the cloud architectures. The researchers used various ML techniques to create a model for predicting the workload. To enhance the model's performance and flexibility, we used a regression-based dataset that was recently updated, which was used with different ML approaches to predict better performance in the cloud architectures. By using the Generalized Linear Model, we achieved the highest performance. The R^2 value refers to the goodness of the model and its performance. Using cloud datasets and machine learning with cloud architectures enhances performance using the different techniques in this paper, resulting in a more generalizable model with overfitting risk. This study focuses on refining the execution of cloud architectures with the help of ML.

Keywords: machine learning; cloud architecture; workload prediction; resource allocation; model performance



Academic Editors: Debopriyo Roy, George F. Fragulis and Peter Ilıc

Published: 9 September 2025

Citation: Asghar, A.; Rehman, A.U.; Ayaz, R.; Suryana, A. Smart Cloud Architectures: The Combination of Machine Learning and Cloud Computing. *Eng. Proc.* **2025**, *107*, 74. <https://doi.org/10.3390/engproc2025107074>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Cloud computing (CC) delivers computing services over the internet and provides resources and access to users without physical infrastructure. Cloud resource management is a process allocating and deallocating resources according to demand so that resources can be used effectively, with minimized cost and high utilization and performance. ML and cloud architectures enhance cloud resource management by predicted analysis, optimized workload, and automatic scaling [1,2]. For instance, supervised learning techniques like Decision Trees (DT), Support Vector Machines (SVM), K Nearest Neighbor (KNN), and

many others evaluate attested and actual-time data that predict workloads, which helps increase performance and utilization. By using ML algorithms to predict the workload of a Google dataset, which has half a million tasks, a model predicted 80% of the cloud workload and improved resource utilization and energy consumption [3]. Computing resources are allocated dynamically by auto-scaling such Virtual Machines (VMs) or storage based on the real-time demand [4]. CC faces several challenges, including efficient allocation of workload complexity and resource cost optimization. Integrating ML into cloud architectures improves decision-making, resource allocation, forecasting demand, and workload distribution [5]. The ML model uses historical data to predict resource needs according to cloud architectures that optimize the cloud network by enhancing service load balancing and workload complexity. Using ML algorithms and cloud architectures to predict consumer requests and workload helps to reduce the cost and traffic of cloud consumers and ensures a high availability of resources. ML-and-cloud architecture combination is a powerful tool for innovation that advances the technology in different features like predictive analysis, resource optimization, creating scalable solutions, making cloud platforms more intelligent and efficient, enhanced security, and seamless scalability.

ML algorithms make cloud architectures smart and accurately predict the needs of resources (CPU, memory, storage) and then allocate dynamically, resolving the issues of over-provisioning and under-utilization. In this research, we integrate the ML of different algorithms in cloud architectures for better performance and increased utilization. ML predicts consumer requests and workload by using workload complexity and services, load-balancing cloud architectures. Workload cloud architectures help to improve the performance of resources and allocate them according to demand, and service load balancing in cloud architectures helps to improve utilization, increase availability, and better performance by using an ML model that predicts the load according to the needs. Our framework seeks to improve the execution of cloud resources by using different techniques and methods that help better model prediction [6]. The researchers applied ML techniques in CC resource management by using supervised, unsupervised, and reinforcement learning to predict current and future workloads. The main objective is to minimize the servers or VMs to keep them safe from power and expenses [7]. Furthermore, some researchers apply ML algorithms to predict security in the cloud for data management to more efficiently solve issues and improve security threats in CC. This research is based on cloud resource performance in cloud architectures that face the issues of workload and performance due to the heavy traffic of cloud consumers, so to achieve high performance and efficiency in CC, we used the cloud resource management dataset from Kaggle. This dataset contains all the effective features that help to refine the utilization, execution, and stability of the data. Figure 1 shows the cloud computing resources. We used several methods like feature selection and a synthetic approach to extract the essential features. Different ML algorithms like Logistic Regression (LR), Gradient Boosting (GBT), Generalized Linear Model (GLM), DT, SVM and KNN were applied to maximize the performance. Figure 2 shows the research paper workflow.

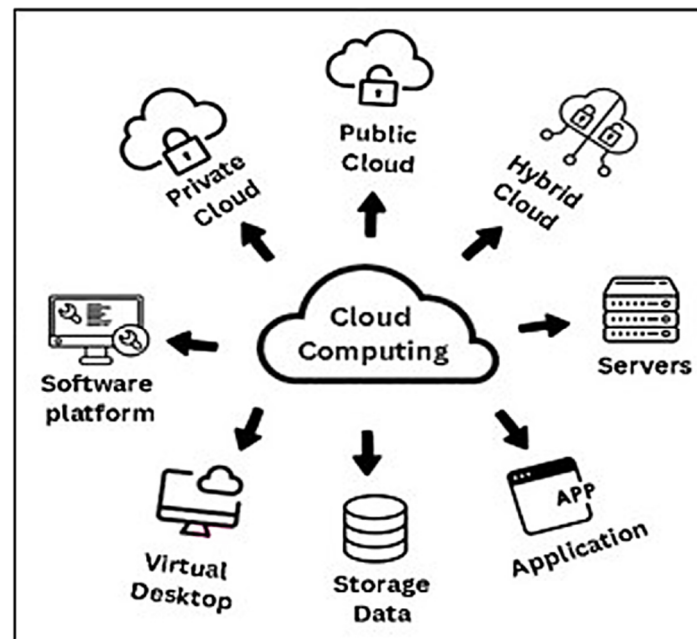


Figure 1. Cloud computing resources.

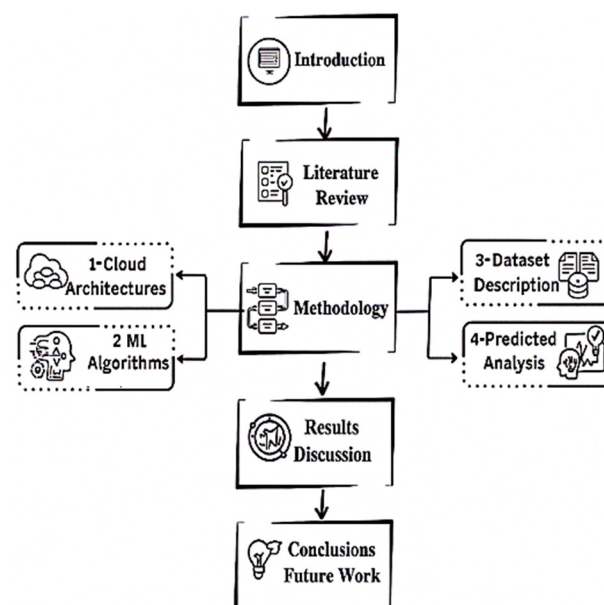


Figure 2. Research paper workflow.

2. Literature Review

Cloud resource management improves resource allocation, costs, and efficiency in cloud computing, and supervised, unsupervised, and reinforcement learning techniques are also used. Ref. [1] surveyed the latest objections to power effectiveness and cloud source control using ensemble learning methods. Machine learning techniques for resource management in CC focus on workload, optimizing costs, energy, and resource utilization [8,9]. Some focus on statistical approaches and popular ways to predict workload. The resource management [10] has some features like task scheduling, virtual machine placement, and VM rescheduling and relocation. Challenges face managing heterogeneous cloud infrastructures and balancing energy, cost, and workload. One issue faced is abnormal workload prediction. Workload prediction for CC is systematic resource optimization and maintains the quality of services by suggesting collection grounded in workload forecast

techniques for excellent perfection of CPU usage and memory usage [11]. ML and big data can handle dynamic workloads and help to improve production in large-scale factories; they use techniques to predict energy usage and resource needs, making factories run evenly [12,13]. ML in cloud architectures helps increase utilization, availability, and performance. Resource managers can easily allocate resources according to the base of prediction. Unsupervised learning and big data processing enhance the system's manageability and result in better real-time decision-making [14]. ML techniques optimize resource usage, improving performance and reducing downtime and decision-making. ML algorithms in dynamic resource allocation for CC, such as supervised learning tasks similar to regression, SVM and Random Forests (RFs), help to forecast source requirements grounded in authentic data [15]. Resource management and workload prediction in CC environments using ML techniques and their application's increasing dependence on virtualization and cloud services make resource allocation better. By using the ML technique for workload prediction using the Mean Gap (M-GAP) forecasting and Zero Gap (0-GAP) forecasting procedures, the enhancement is between 75% and 90% in CPU use forecast and 92% to 95% in memory use forecast. Applications of workload prediction methods such as neural networks and SVM have focused on improving prediction accuracy with clustering strategies such as prediction clustering and dynamic-based clustering to provide magnified results in cloud resource management, and ML approaches have crucially enabled better prediction, scheduling, and resource allocation [16]. Future strategies will focus on improved system manageability and elasticity, multi-dimensional resource management, and the integration of sophisticated learning techniques [17,18] to meet the demands of quickly expanding cloud applications. Workload prediction approaches like neural networks and SVM have focused on making prediction accuracy better using clustering techniques like prediction clustering and dynamic-based clustering to enhance cloud resource management results [19,20]. Better scheduling, prediction, and resource allocation have also been made possible by machine learning techniques [21]. Importantly, prediction, scheduling, and resource allocation have increased.

3. Methodology

The proposed methodology for this research uses CC architectures integrated with ML to improve performance of cloud resources, where it can manage resources and distribute workloads using the cloud architectures. The performance predictions are made by using ML algorithms such as LR, GBT, GLM, DT, SVM, PR, and KNN. The cloud architectures are the workload architecture and service load balancing. By using the architectures to predict the performance of cloud resources, we use regression data to improve the accuracy and execution of workload prediction. Overall, the key point as to methodology deals with creating an effective model using ML to predict the resource workload based on architectures.

3.1. ML Enhanced Cloud Architectures

3.1.1. Workload Architecture

The workload architecture handles the available resources to optimize performance and prevent disturbance. According to the workload complexity dataset features, initial utilization and initial performance scores are inputs to predict performance improvement using ML models. ML predicts the current and future workload based on historical data and ensures that the resources are dynamically allocated and reallocated according to their needs for better efficiency, as shown in Figure 3.

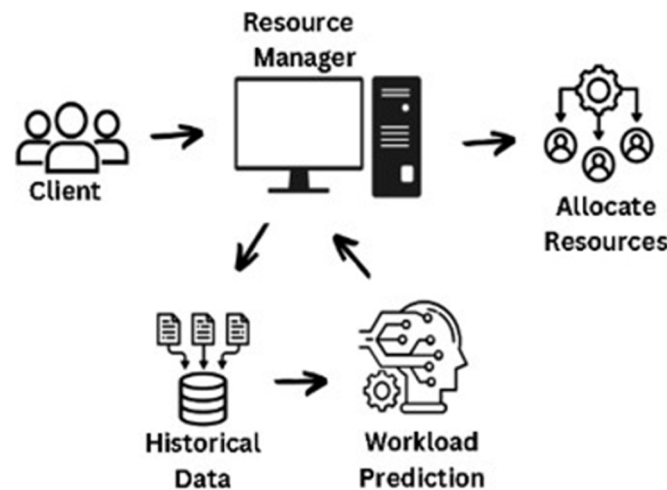


Figure 3. Workload prediction.

3.1.2. Service Load Balancing Architecture

Load balancing distributes the incoming network traffic and workload from multiple servers or resources in CC to ensure high availability and performance. ML models can help minimize under-provisioning or over-provisioning which leads to a balance between cost and performance. The load balancer distributes workload according to the model traffic prediction and more efficiently reduces suspension, and faster response time can improve performance or help prevent bottlenecks, as shown in Figure 4.

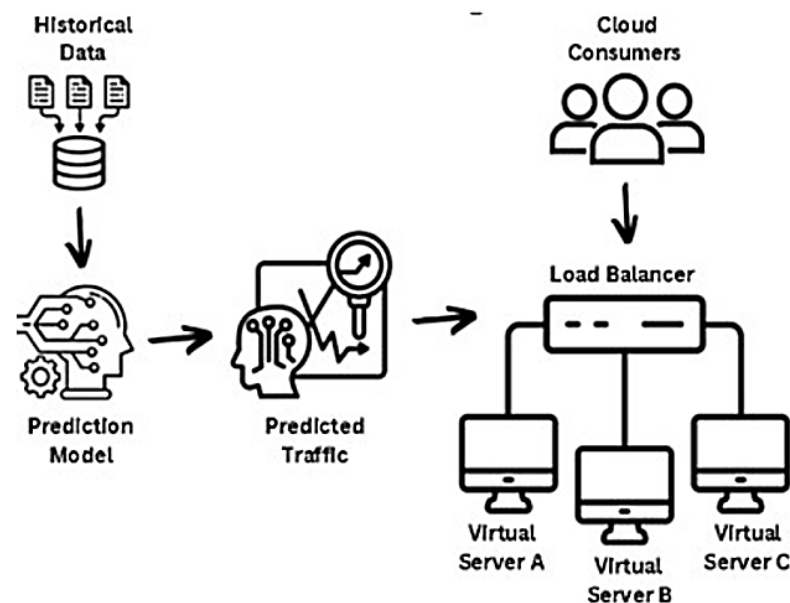


Figure 4. Service load prediction.

3.2. Machine Learning Classifiers

3.2.1. Linear Regression

LR exists as a supervised machine learning classifier used for continual outcomes; in our case, linear regression utilizes the model relationship between features like the initial resource pool, initial cost per unit, and workload complexity to predict performance improvement. It pretends to relate to I/O features. The simpleness of linear regression makes it interpretable and offers a perception of resource allocation and utilization efficiency directly to performance outcomes.

3.2.2. Gradient Boosted Tree

GBT is an ensemble learning algorithm that builds models according to a sequence; each model corrects the errors in the previous one. We used GBT to predict performance improvement through features like optimized resource allocation, initial performance score, and cost efficiency improvement. GBT handled the missing data and feature importance that helped us to understand which ingredient had the greatest influence on performance improvement.

3.2.3. K-Nearest Neighbor

The KNN classifier exists as a supervised machine learning model that is applied for both regression and classification problems. KNN predicts and is dependent on the nearest data end in a special space. “K” in KNN is the digits of neighbors. In our case, we used KNN to predict performance based on features’ initial resource pools, initial cost, and workload complexity. It is used for smaller datasets because of its simplicity and minimal training. KNN has distinct ways to measure the distance between points. Distance metrics are very important because KNN accuracy depends on distance.

3.2.4. Decision Tree

DT exists as a supervised ML classifier that is applied for classification problems or regression problems. The algorithm aids in predicting improved performance based on features’ initial performance score and optimizes resource allocation and initial utilization. The tree has structured nodes and branches, and the leaf nodes represent the prediction value or class label. Its Greedy approach is the step where it splits and obtains the best possible data like Gini impurity and Entropy. It is an easy method because every decision is clear and visual.

3.2.5. Generalized Linear Model

A GLM is a model that is used for regression and classification and uses a wide range of relationships among predictor and response variables when the response variables do not track the usual dispensation. Features like initial performance score, optimized resource allocation, and cost efficiency improvement can predict improved performance.

3.2.6. Support Vector Machine

SVM exists as a supervised ML classifier applied to classification problems and regression problems. SVM finds the best boundary that disconnects the ends of separate classes. SVM forecasts the improved performance finds the hyperplane and separates different points. The edge is the distance between the plane and closest support vectors. We used features like the initial resource pool, workload complexity, and optimized performance score. SVM’s main point is to increase the edge for exceeding generalization. SVM can manage linear and non-linear data using kernel tricks.

3.3. Machine Learning Framework

Our framework is the guidance, tools, and libraries that are used to build, develop, and implement ML models. The framework also works with several datasets and ML algorithms, to make development faster. The framework supports several ML techniques, including supervised and unsupervised learning, and can integrate with cloud services. It has a sequence of processes, including data collection, model selection, training, model evaluation, and deployment. In Figure 5, we present a dataset from Kaggle that is updated. After collecting data, we started data preprocessing, cleaned the data, and replaced missing values. After cleaning the data, we used the outlier removal approach to balance the data and variables and chose to minimize the number of inputs by picking the most

relevant variables. Then the data was divided into developing, validating, and testing parts. In developing and validation, we selected the input features and target features; after selecting features, we selected the ML algorithms according to the dataset. In the ML algorithms' processing of the model predictions, some values were used in formulas to examine the performance of the model; end results are displayed in the table at the end of the discussion below.

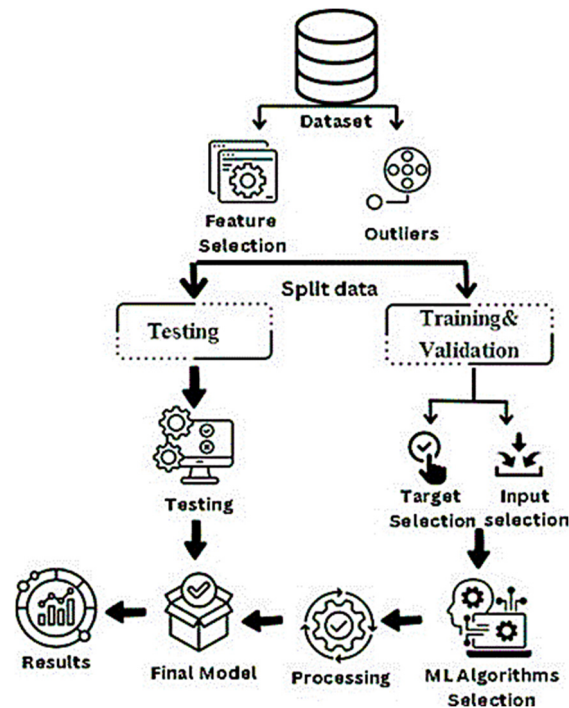


Figure 5. Machine Learning Framework.

3.3.1. Feature Selection

The procedure of picking out and appointing the highest applicable variables and removing the non-applicable variables improved the model's accuracy and performance.

Feature selection was applied to predict improved performance. It had many procedures for variable selection like the refining process, envelope process, and encased process. We combined features like workload complexity and initial utilization to devise novel variable workload utilization and understand the connection among the resources and workload, which was used for bettering performance and leading to accurate prediction.

3.3.2. Outlier

An outlier is a data point that lies at an abnormal distance in a random sample from a population. Outliers can impact model accuracy, which makes their identification crucial. The tool Rapid Miner can be applied to outliers to detect or generate outlier scores which helps to handle outliers and make the accuracy of the model the best.

3.3.3. Split Data

The data splitting process divided the dataset into separate sets such as training and validation and testing. The training and validation sets were employed to develop the ML standard and make predictions, and the authentication data was employed to evaluate performance during training to ensure that it did not overfit. Finally, testing data was used to evaluate model performance on unseen data after training was completed. The data was split into 70% for training and validation and 30% for testing to ensure that the model avoided underfitting and overfitting on unseen data.

3.4. Tool

RapidMiner is an open-source platform popular in data science and ML tools. It is used for data preparation, modeling, evaluation, and deployment. We used RapidMiner 10.3 to predict CC resource management.

3.5. Dataset Description

There are many datasets on Kaggle related to CC resource management. Still, we used an updated dataset that is a CC resource prediction dataset based on the basics of its 12 features. The dataset has various parameters of resource allocation both before and after optimization. The number of instances in the dataset is 1000. The target features are utilization improvement, cost efficiency improvement, and performance improvement. The features and their descriptions are shown in Table 1.

Table 1. Features and their descriptions.

Feature	Description
Initial resource pool	The set of computing resources available for cloud services before workload execution begins.
Workload complexity	How demanding or complicated a task being processed is.
Initial utilization	The available resource pool being used for the workload execution.
Initial cost per unit	The initial cost of the resources before the execution.
Initial performance score	A score of the workload before optimization.
Optimized resource allocation	Dynamic reallocation of resources based on the actual demand of the task.
Optimized utilization	The percentage of resources used after the optimization process.
Optimized cost per unit	The cost of used resources after the optimization.

3.6. Predicted Analysis

The ML algorithm was selected to find the best execution with the dataset. We compared to the predicted value the actual outcomes for the best performance and reliability. So, we applied various ML algorithms [22–24] such as LR, GBT, GLM, DT, SVM, PR, and KNN. According to the framework, all algorithms were applied individually to check the performance and obtain a higher R^2 , which meant that the model had high goodness and performed very well. The RMSE and MAE explain prediction errors but they are sensitive to outliers; both values are better when they are lower [25]. Many classifiers were applied to the dataset, which is described above in this paper, but the best one which fitted the model and gave the best results is described below.

3.7. Predictive Modeling Process

The GLM uses linear regression to handle several types of data distributions, not just normal distributions. The GLM achieved the best R^2 value of 1.000 and the model explained the goodness of the variance in the data; also, the RMSE and MAE showed minimal prediction errors. So, the GLM predicted performance accurately and was reliable choice for the dataset [26,27]. RapidMiner was used to check the best result. In the processing, we selected a set role to describe the role and then selected an attribute from the dataset; after this, we split the data and used the GLM. The RapidMiner processing for the GLM is shown in Figure 6.

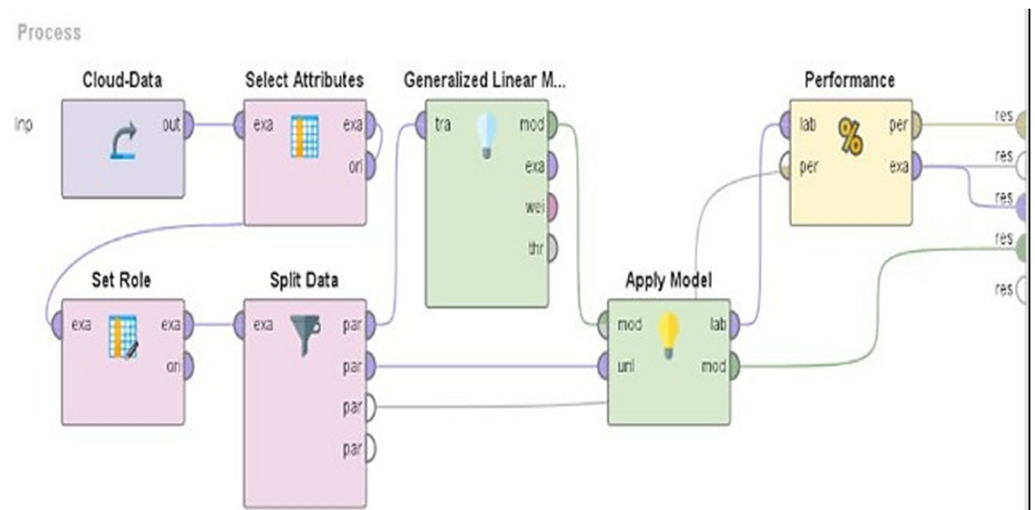


Figure 6. GLM.

4. Results

This paper discusses the findings of the predicted analysis of the performance enhancement in CC by ML. The results are based on different ML algorithms using cloud architectures to achieve better performance. According to our dataset, regression data accuracy was found through evaluation metrics. Commonly used methods for this are R^2 , RMSE, and MAE. These results are shown in Table 2. R^2 is the proportion of variance explained by the model; it being closer to 1 means better model performance. The combination of R^2 and RMSE to find the overall accuracy for regression data is used because it covers both the model fit and error. The table below shows the accuracy found based on R^2 . The values are given below. Given the data, it can be noticed that the GLM is the winning model, as it has the maximum R^2 value of 1.000, the lowest RMSE of 0.179, and an almost low MAE of 0.086, but Decision Tree is better than the GLM. Decision Tree has an R^2 value of 0.999, which is quite good, RMSE of 0.132, and MAE of 0.080, which is better than that of the GLM. LR comes in third with a satisfactory R^2 of 0.996, with a borderline RMSE of 0.488 and MAE of 0.295. Other models such as SVM, PR, and KNN have a lower R^2 and wider scope of errors and, therefore, performed poorly. The given chart compares the ML classifiers applied to improve the performance prediction, with the R^2 achieved by cloud resource workload prediction. For comparison, Table 3 shows previous work performed on the proposed system.

Table 2. Performance metrics of different algorithms.

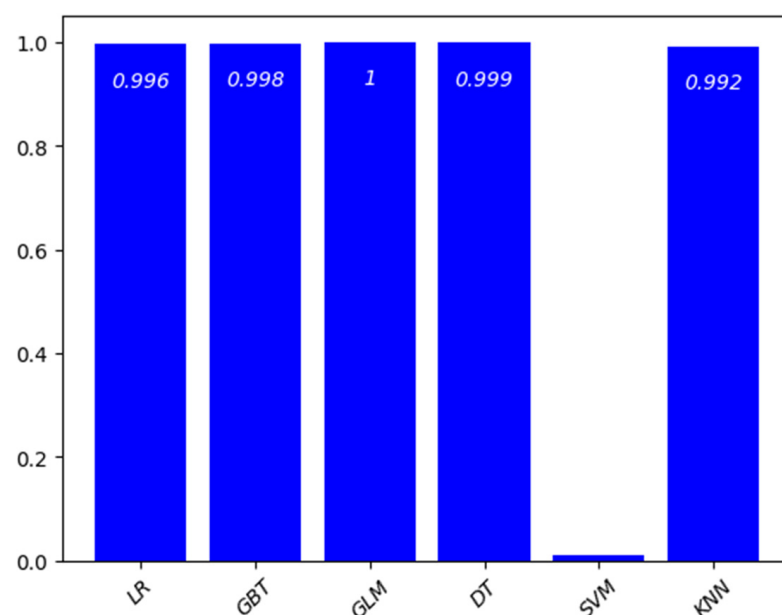
Algorithm	R^2	RMSE	MAE
LR	0.996	0.488	0.295
GBT	0.998	0.523	0.299
GLM	1.000	0.179	0.086
DT	0.999	0.132	0.080
SVM	0.000	0.542	0.252
KNN	0.992	0.5671	0.495

Table 3. Comparison of techniques and classifiers with reported accuracy.

Author	Year	Technique	Classifier	Accuracy
IKHLASSE [18]	2020	CGPANN	Neural Network	97.81%
FATIMA [19]	2019	NOMA	SVM	98%
THANG [20]	2019	QOS	Auto-scaling	91%
GAITH [21]	2020	RNN-LSTM	DL	96%
JIECHAO [22]	2020	T-D	Clustering-based	95%
SUKHPAL [23]	2019	HRM	CO	95%
UMER [24]	2020	DDoS	SVM	99.7%

5. Conclusions

This research used ML algorithms in cloud architectures due to issues and challenges faced related to workload complexity and the traffic of cloud consumers. Using ML algorithms, that is, LR, GBT, GLM, DT, SVM, PR, and KNN, in the workload cloud architecture and service load balancing cloud architecture can improve the performance and decrease the workload according to dynamic allocation. Among all the classifiers, the GLM had the maximum R^2 value which means that the model had goodness of fit. The GLM achieved the best result using the cloud resources to increase the performance. The framework was used for all classifiers which divided data which were for testing or training and validation. Input data were selected and target data were selected for training and testing. After the data selection, the model was selected for an ML algorithm; all algorithms succeeded using the framework and the best performance was gained. After the testing, the result was checked by using different formulas and the best result was given by the GLM classifier. The testing model used in this paper for machine learning was RapidMiner; through RapidMiner, the result was obtained and it was decided which was the best for the cloud resource performance. The ML model can predict real-time workload and dynamically allocate and deallocate cloud resources, which optimizes cost and reduces resource wastage which helps cloud services to work efficiently. In the future, cloud systems will use the ML model for self-healing capability, modeling cloud logs, and metric analysis that will predict failures and automatically perform the resolution actions. So, downtime will be minimized and the system will be reliable. The results are visualized in Figure 7.

**Figure 7.** Classifiers' R^2 comparison.

Author Contributions: Conceptualization, A.A. and A.U.R.; methodology, A.A. and R.A.; software, A.A.; validation, A.A., A.U.R. and R.A.; formal analysis, A.A. and R.A.; investigation, A.A. and R.A.; resources, R.A.; data curation, A.A. and R.A.; writing, original draft preparation, A.A.; writing, review and editing, A.U.R., R.A. and A.S.; visualization, A.A. and R.A.; supervision, A.U.R. and A.S.; project administration, A.U.R.; funding acquisition, A.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data supporting the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hamzaoui, I.; Duthil, B.; Courboulay, V.; Medromi, H. A Survey on the Current Challenges of Energy-Efficient Cloud Resources Management. *SN Comput. Sci.* **2020**, *1*, 2. [\[CrossRef\]](#)
2. Zhang, R. The Impacts of Cloud Computing Architecture on Cloud Service Performance. *J. Comput. Inf. Syst.* **2020**, *60*, 166–174. [\[CrossRef\]](#)
3. Hussain, F.; Hassan, S.A.; Hussain, R.; Hossain, E. Machine Learning for Resource Management in Cellular and IoT Networks: Potentials, Current Solutions, and Open Challenges. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 1251–1275. [\[CrossRef\]](#)
4. Yadav, M.P.; Rohit; Yadav, D.K. Resource Provisioning Through Machine Learning in Cloud Services. *Arab. J. Sci. Eng.* **2022**, *47*, 1483–1505. [\[CrossRef\]](#)
5. Kumar, P.; Kumar, R. Issues and Challenges of Load Balancing Techniques in Cloud Computing: A Survey. *ACM Comput. Surv.* **2019**, *51*, 6. [\[CrossRef\]](#)
6. Dritsas, E.; Trigka, M. A survey on the applications of cloud computing in the industrial internet of things. *Big Data Cogn. Comput.* **2025**, *9*, 44. [\[CrossRef\]](#)
7. Goodarzy, S.; Nazari, M.; Han, R.; Keller, E.; Rozner, E. Resource Management in Cloud Computing Using Machine Learning: A Survey. In Proceedings of the 19th IEEE International Conference on Machine Learning and Applications (ICMLA 2020), Miami, FL, USA, 14–17 December 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 811–816. [\[CrossRef\]](#)
8. Jeyaraman, J.; Bayani, S.V.; Malaiyappan, J.N.A. Optimizing Resource Allocation in Cloud Computing Using Machine Learning. *Eur. J. Technol.* **2024**, *8*, 12–22. [\[CrossRef\]](#)
9. Wang, Y.; Bao, Q.; Wang, J.; Su, G.; Xu, X. Cloud Computing for Large-Scale Resource Computation and Storage in Machine Learning. *J. Theory Pract. Eng. Sci.* **2024**, *4*, 163–171. [\[CrossRef\]](#)
10. Wang, Y.; Zhu, M.; Yuan, J.; Wang, G.; Zhou, H. The Intelligent Prediction and Assessment of Financial Information Risk in the Cloud Computing Model. *arXiv* **2024**, arXiv:2404.09322. [\[CrossRef\]](#)
11. Shafiq, D.A.; Jhanjhi, N.Z.; Abdullah, A. Load Balancing Techniques in Cloud Computing Environment: A Review. *J. King Saud Univ. Comput. Inf. Sci.* **2022**, *34*, 3910–3933. [\[CrossRef\]](#)
12. Patel, J.; Jindal, V.; Yen, I.L.; Bastani, F.; Xu, J.; Garraghan, P. Workload Estimation for Improving Resource Management Decisions in the Cloud. In Proceedings of the 12th IEEE International Symposium on Autonomous Decentralized Systems (ISADS 2015), Taichung, Taiwan, 25–27 March 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 25–32. [\[CrossRef\]](#)
13. Moreno-Vozmediano, R.; Montero, R.S.; Huedo, E.; Llorente, I.M. Efficient Resource Provisioning for Elastic Cloud Services Based on Machine Learning Techniques. *J. Cloud Comput.* **2019**, *8*, 1. [\[CrossRef\]](#)
14. Gill, S.S.; Garraghan, P.; Stankovski, V.; Casale, G.; Thulasiram, R.K.; Ghosh, S.K.; Ramamohanarao, K.; Buyya, R. Holistic Resource Management for Sustainable and Reliable Cloud Computing: An Innovative Solution to Global Challenge. *J. Syst. Softw.* **2019**, *157*, 110395. [\[CrossRef\]](#)
15. Bi, J.; Li, S.; Yuan, H.; Zhou, M.C. Integrated Deep Learning Method for Workload and Resource Prediction in Cloud Systems. *Neurocomputing* **2021**, *424*, 35–48. [\[CrossRef\]](#)
16. Mijuskovic, A.; Chiumento, A.; Bemthuis, R.; Aldea, A.; Havinga, P. Resource Management Techniques for Cloud/Fog and Edge Computing: An Evaluation Framework and Classification. *Sensors* **2021**, *21*, 1832. [\[CrossRef\]](#) [\[PubMed\]](#)
17. García, Á.L.; De Lucas, J.M.; Antonacci, M.; Zu Castell, W.; David, M.; Hardt, M.; Iglesias, L.L.; Moltó, G.; Plociennik, M.; Tran, V.; et al. A Cloud-Based Framework for Machine Learning Workloads and Applications. *IEEE Access* **2020**, *8*, 18681–18692. [\[CrossRef\]](#)

18. Ikram, H.; Fiza, I.; Ashraf, H.; Ray, S.K.; Ashfaq, F. Efficient Cluster-Based Routing Protocol. In Proceedings of the 3rd International Conference on Mathematical Modeling and Computational Science: ICMCMC, Chandigarh, India, 11–12 August 2023.
19. Althati, C. Machine Learning Solutions for Data Migration to Cloud: Addressing Complexity, Security, and Performance. Available online: <https://sydneyacademics.com/> (accessed on 11 November 2024).
20. Prasad, V.K.; Bhavsar, M.D. Monitoring and Prediction of SLA for IoT Based Cloud. *Scalable Comput.* **2020**, *21*, 349–357. [[CrossRef](#)]
21. Tatineni, S.; Chakilam, N.V. Integrating Artificial Intelligence with DevOps for Intelligent Infrastructure Management: Optimizing Resource Allocation and Performance in Cloud-Native Applications. *J. Bioinform. Artif. Intell.* **2024**, *4*, 109–142.
22. Botvich, A. Machine Learning for Resource Provisioning in Cloud Environments. In Proceedings of the 2020 IEEE International Conference on Cloud Engineering (ICEE), Sydney, Australia, 10 January 2020.
23. Moghaddam, S.K.; Buyya, R.; Ramamohanarao, K. Performance-Aware Management of Cloud Resources: A Taxonomy and Future Directions. *ACM Comput. Surv.* **2019**, *52*, 4. [[CrossRef](#)]
24. Hong, C.H.; Varghese, B. Resource Management in Fog/Edge Computing: A Survey on Architectures, Infrastructure, and Algorithms. *ACM Comput. Surv.* **2019**, *52*, 5. [[CrossRef](#)]
25. Lim, M.; Abdullah, A.; Jhanjhi, N.; Khurram Khan, M.; Supramaniam, M. Link prediction in time-evolving criminal network with deep reinforcement learning technique. *IEEE Access* **2019**, *7*, 184797–184807. [[CrossRef](#)]
26. Diwaker, C.; Tomar, P.; Solanki, A.; Nayyar, A.; Jhanjhi, N.; Abdullah, A.; Supramaniam, M. A New Model for Predicting Component-Based Software Reliability Using Soft Computing. *IEEE Access* **2019**, *7*, 147191–147203. [[CrossRef](#)]
27. Airehrour, D.; Gutierrez, J.; Kumar Ray, S. GradeTrust: A secure trust based routing protocol for MANETs. In Proceedings of the 2015 25th International Telecommunication Networks And Applications Conference (ITNAC), Sydney, Australia, 18–20 November 2015; pp. 65–70.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.