International Journal of Emerging Research in Engineering and Technology
Pearl Blue Research Group| Volume 6 Issue 4 PP 27-37, 2025
ISSN: 3050-922X | https://doi.org/10.63282/3050-922X.IJERET-V6I4P104

Original Article

Data-Driven Cloud Workload Optimization Using Machine Learning Modeling for Proactive Resource Management

Jeremiah Emavwohwe Kofi
Senior Multi-Cloud Architect, Centrica PLC.

Received On: 19/08/2025 ***Revised On: 30/08/2025*** ***Accepted On: 24/09/2025*** ***Published On: 12/10/2025***

Abstract - *Cloud computing has reinvented delivery of services due to its provision of scalability, elasticity and cost effectiveness. But increased workloads have aggravated management of resources, which has resulted in energy waste, SLA break, and high cost of operation. The study presents an active data-driven optimization model based on machine learning to predict workload dynamics. Google Cluster Trace dataset is processed using logarithmic scaling, Savitzky-Golay filtering and Min-max normalization in order to improve stability and quality. An LSTM model is used to find longitudinal dependencies and forecast workload, CPU, and RAM usage. Measures of evaluation are R^2, Mean Squared error (MSE) and Root mean Squared logarithmic error (RMSLE). The findings indicate that LSTM attains almost perfect accuracy, R^2 of 0.99%, MSE of 13,934.54 (workload), 128.89 (CPU) and 131.29 (RAM), and RMSLE of 0.15, 0.16 and 0.14. Compared to existing models such as VAMBig, SVM, and SATCN, the LSTM framework significantly outperforms them in prediction accuracy and error minimization. These findings confirm that proactive LSTM-based workload optimization reduces energy usage, enhances SLA compliance, and strengthens scalability in dynamic cloud computing environments. Future work can extend this framework to multi-cloud and federated environments for broader applicability.*

Keywords - *Cloud Workload Optimization, Proactive Resource Management, Machine Learning, Resource Allocation, Time Series Modeling, Cloud Computing.*

1. Introduction

Cloud computing has just become a paradigm of hosting and providing Internet-based services. It is highly attractive to businesses because it is flexible and they pay as they go since the business does not have to plan to supply its resources ahead of time and only increase resources when there is an influx in demand [1][2]. The rapid growth in the use of cloud computing has led to wide spectrum of infrastructure, platform and software level of services. Nevertheless, this growth has its own issues especially in the field of resource consumption [3]. The current generation data center cloud infrastructure has large power demands, which result into high cost of operations and carbon footprint. Meanwhile, most of these data centers are not at optimal levels of utilization, with a lot of power going to waste because of over-provisioned and idle resources [4][5].

The importance of cloud workload optimization by these inefficiencies can be seen to guarantee cloud processing that is energy efficient, cost-reduction and sustainable computing infrastructures. Cloud workload optimization is an internal process to assign tasks to the virtualized infrastructure in a strategic manner to fulfill such goals as performance, scale, and cost-effectiveness. However enterprise workloads are non-uniform and dynamic in nature [6][7][8]. The high business activity of the season, unpredictive user demands, multi-tenancy setups, and a variety of service-level agreements present huge complexities. Traditional models of optimization, like rule-based autoscaling, heuristic scheduling and resource allocation based on a set of thresholds are not capable of dealing with this unpredictability. These strategies are also prone to wasting resources, violation of SLA, or poor performance when the instances of work load peaks are not forecasted [9][10]. This, in turn, makes the necessity to make real-time decisions under smart, highly adaptable and data-driven optimization procedures very pressing. The solutions should not merely address the fluctuations but should additionally offer proactive and sustainable work load control in large scale cloud environment.

Machine learning (ML) has emerged as one of the significant enablement of smart workload optimization in cloud computing [11]. Unlike the other methods of the system, ML allows the system to learn using historical as well as real-time information and adapt to the dynamism of the environment as well as enhance over time [12]. With the help of the ML techniques, predictive models on the demand of resources can be made, anomalies in the workload pattern can be detected, and resource scaling can be made dynamically responsive to the changes [13][14]. In comparison with the traditional methods, ML models are more likely to forecast utilization patterns, automatically schedule the work, and allocate it more efficiently to distributed infrastructure due to large amounts of data analysis [15][16]. Workload management proactively, unlike the reactive strategies, which respond to issues after they arise, allows the management of the workload of enterprises with scalability and broad applicability to cost-effective and sustainable cloud operations.

1.1. Motivation and Contribution of Paper

The study is driven by the fact that cloud computing environments are increasingly becoming complex with dynamically changing workloads and unpredictably changing resource demands. Inefficiency in the manner in which resources are distributed can be the cause of the high operating costs, wastage of energy and poor quality of services. The conventional method of reaction is not adaptable enough to act to a sudden surge of workload or instances of underutilization. This presents a tremendous requirement on active and knowledge-based solutions that have the capacity to effectively predict the workload trends and help in managing its resources within an opportune and optimized manner. On the basis of the big data such as the Google Cluster Trace, researchers have been able to acquire the real-world patterns of workloads that give a foundation on predictive modeling to enhance the scalability, efficiency, and reliability within the cloud infrastructure. The major study findings are the following:

- Utilized the Google Cluster Trace dataset as a realistic source of workload data.
- Applied essential preprocessing techniques, including logarithmic scaling, Savitzky–Golay filtering, and Min–Max scaling, to prepare the data for modeling.
- Developed and implemented an LSTM-based predictive framework for modeling workload dynamics.
- Evaluated model performance using R², MSE, and RMSLE to ensure comprehensive assessment of accuracy and error distribution.
- Provided insights into proactive resource management strategies for cloud environments.

1.2. Justification and Novelty of the Paper

The justification for this study lies in the growing complexity of cloud environments, where traditional resource management strategies struggle to adapt to unpredictable workload fluctuations. Inefficient allocation contributes to energy wastage, SLA violations, and escalating operational costs, necessitating data-driven, proactive approaches. Integrating advanced preprocessing methods, LSTM modelling for workload prediction, and real-world Google Cluster Trace data is what makes this article unique. Unlike prior heuristic or static optimization methods, the proposed framework ensures adaptability, high predictive accuracy, and practical applicability, offering a scalable solution for efficient, autonomous cloud workload management.

1.3. Structure of the Paper

The following is the outline of the paper: Related works are included in Section II. The technique, including the description of the dataset, preprocessing, and model implementation, is detailed in Section III. In Section IV, review the performance measures and talk about the findings of the experiments. Section V summarizes the work and suggests avenues for further investigation.

2. Literature Review

The reviewed studies apply machine learning and AI techniques for cloud workload optimization, combining prediction, migration, and scheduling to proactively enhance resource utilization, minimize energy consumption, and improve SLA adherence in cloud environments. Xin (2025) highlights the significant energy consumption of cloud data centers and proposes an AI-driven approach using deep reinforcement learning (RL) to optimize workload migration decisions. This approach, which was trained using publicly available cloud workload traces, minimizes service-level agreement (SLA) violations and reduces energy consumption by more than 20% compared to baseline heuristics. By reducing the number of servers needed to run applications, this method improves resource utilization and reduces energy usage [17].

Karthikeyan et al. (2025) propose a novel approach for workload prediction in cloud data centers, addressing the low accuracy of existing methods due to redundancy, noise, and low accuracy. Compared to the current methods, the CVSTGCN-WLP-CDC method yields a better accuracy of 23.32, 28.53, and 24.65, consumes less energy of 22.34, 25.62, and 22.84, respectively. This new strategy has provided a more effective and precise workload prediction tool within the cloud information centers [18]. Diwaker and Miglani (2024) have introduced an adaptive workload forecaster based on Particle Swarm Optimization-enhanced autoencoder to forecast upcoming workloads in cloud data centers. The model relies on historical data and it is better than conventional methods because it decreases the scores of RMSE and MAE by about 75 percent and 76 percent respectively. Another major successful outcome of the model is the impressive increase in the R2 score to the value close to 0.96 percent which is a strong workload prediction solution to the dynamic cloud environment. The method is suitable to solve the issue with dynamically changing resource requirements and redundant customer requests, as it provides precise forecasts in real-time conditions [19].

Ali et al. (2024) suggested a hybrid system that integrates deep learning models with evolutionary algorithms in predicting workload in cloud computing. Predicting the future workloads, the model is a neural network optimization of the differentiation evolution, which is based on a new mutation strategy. The model was applied to the real-life traces of Google and the Alibaba platform and the results indicated a low error rate of 0.0002. High accuracy and automaticity characteristics of the model have possible applications in cloud computing including real-time applications. The findings have been correlated to the available literature, which show the possibilities of this model in the cloud computing [20]. Ahamed et al. (2023) present a new solution, Federated Cloud

Workload Prediction with Deep Q-Learning (FEDQWP), to the complicated VM placement issue, power efficiency, and SLA conservation in Federated Cloud Computing (FCC) settings. The FEDQWP model also takes advantage of deep learning abilities to identify patterns in the background and allocate resources to optimize. The findings indicate that the QLearning model is an efficient model in terms of the use of the CPU at an average of 29.02, the time to complete migrations is an average of 0.31 units, the amount of tasks completed is 699 at an average, the amount of energy used is lowest at an average of 1.85 kWh, and the number of SLA violations is lowest at an average of 0.03 violations proportionally [21].

Dogani et al. (2023) suggest a multivariate time series method to predict workload of host machines in the cloud data centers. The approach involves a statistical analysis to build the training set, a convolutional neural network (CNN) to collect the hidden spatial features among all correlated variables, as well as a GRU network that is optimized by the attention mechanism to acquire temporal correlation features. The accuracy of prediction offered by the proposed method has been enhanced by 2 to 28 percent in comparison with the base methods and past studies. The study facilitate the optimization of resource distribution and prevent service-level agreement alteration (SLA) in cloud computing applications [22].

The Table 1 presents research on cloud workload optimization, methods, data, results, and limitations, and directions toward proactive, efficient, and resource management of SLA.

Table 1: Literature Review of Machine Learning Approaches for Cloud Workload Optimization and Proactive Resource Management

Author(s)	Methodology	Dataset	Key Findings	Limitations	Future Work
Xin (2025)	Deep Reinforcement Learning for workload migration optimization	Public cloud workload traces	Reduced energy consumption by >20% and SLA violations; RL autonomously learns when and where to migrate workloads	Focuses on migration decisions only; performance overhead of live migration not deeply addressed	Incorporate multi-objective optimization considering multiple QoS metrics; real-world deployment validation
Karthikeyan et al. (2025)	The CVSTGCN + Gazelle Optimization Algorithm is a Convolutional Neural Network for Complex-Valued Spatial-Temporal Graphs.	NASA and Saskatchewan HTTP traces	Accuracy improvement up to 28.53%; energy consumption reduced up to 25.62% compared to baselines	Computational complexity; scalability to larger datasets not discussed	Extend to online real-time prediction; integrate proactive resource allocation and scaling
Diwaker and Miglani (2024)	Particle Swarm Optimization (PSO)–enhanced autoencoder for adaptive workload forecasting	Bitbrains dataset	Reduced RMSE and MAE by ~75%; R² score ~0.96, outperforming traditional approaches	Limited to historical datasets; may not adapt to extreme workload spikes in real time	Combine with hybrid optimization for real-time proactive scaling; test in multi-cloud environments
Ali et al., (2024)	DE-NN, an innovative clustering-based mutation approach for evolutionary optimization; clustering-based workload prediction	Google real-world traces, Alibaba platform	Achieved very low RMSE (0.0002) for CPU, RAM, BW prediction; high accuracy validated with R², mean bias, 90th percentile score	Focused on prediction only; real-time deployment and integration with resource management not evaluated	Extend to proactive autoscaling, dynamic resource allocation, and real-time cloud management; explore multi-cloud scenarios
Ahamed et al. (2023)	Deep Q-Learning (DQL) for Federated Cloud Workload Prediction (FEDQWP)	Real-world FCC workloads	Improved CPU utilization (29.02 median), migration time (0.31), finished tasks (699), energy (1.85 kWh), SLA violations (0.03)	Focused on FCC scenario; generalization to multi-cloud or non-federated environments not shown	Incorporate predictive workload forecasting for proactive resource allocation; optimize multi-cloud deployments
Dogani et al. (2023)	Hybrid CNN + GRU with Attention for multivariate multi-step	Google cluster data	Accuracy improved 2–28% over baselines; captures spatial and	Limited to prediction; does not address proactive	Integrate with resource allocation and autoscaling

	workload prediction		temporal correlations effectively	migration, scheduling, or energy optimization	mechanisms for proactive management

3. Methodology

The methodology aims at providing an efficient framework of proactive cloud workload optimization with the help of machine learning indicated in Figure 1. It focuses on the workload dynamics and forecasts the resource requirements to have increased efficiency and reduced overheads in a large-scale cloud setting. To do this, the primary source of data is the Google Cluster Trace dataset which is used to arrive at the real-world workload traces to be used as the model development. The data is subjected to preprocessing such as logarithmic scaling of the data to stabilize the variance, Savitzky-Golay filtering to remove the noise, and Min-Max scaling to normalize the features. The data is preprocessed, and it is then divided into training and testing parts to avoid bias in evaluating the performance of the model. The implemented A Long Short-Term Memory (LSTM) model serves well to deal with sequential dependencies of workload patterns. R², Mean Squared error (MSE) and Root mean squared logarithmic error (RMSLE) are used to evaluate the model since they are used to identify both the accuracy and error behavior. The collected final results prove the applicability of the model to preemptive cloud resource management.

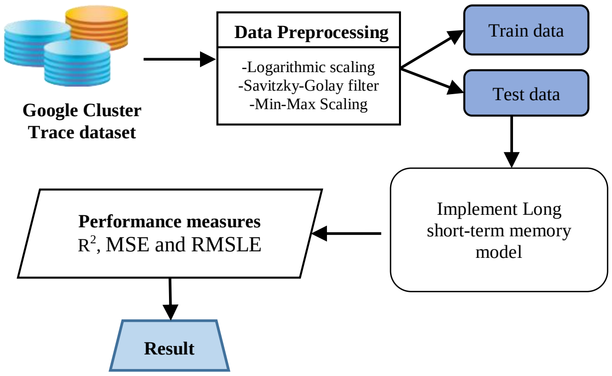

Fig 1: Methodology for Cloud Workload Optimization Using LSTM Model

3.1. Data Collection

The released in 2011 Google cluster workload and resource data has become a popular benchmark in the assessment of prediction methods. It comprises 25,462,157 tasks and 672,004 jobs that were gathered by around 12,000 machines within 29 days. Workload comes in the form of jobs that have several tasks, and the records regarding resource consumption are connected to CPU and RAM usage. The data is categorized into workload and resource utilization time series of 2-minute time slots, which make up 20,880 time slots. Failures and anomalies give these series non-linear, non-stationary and noisy patterns, which become difficult to predict but extremely important to running cloud data centers effectively.

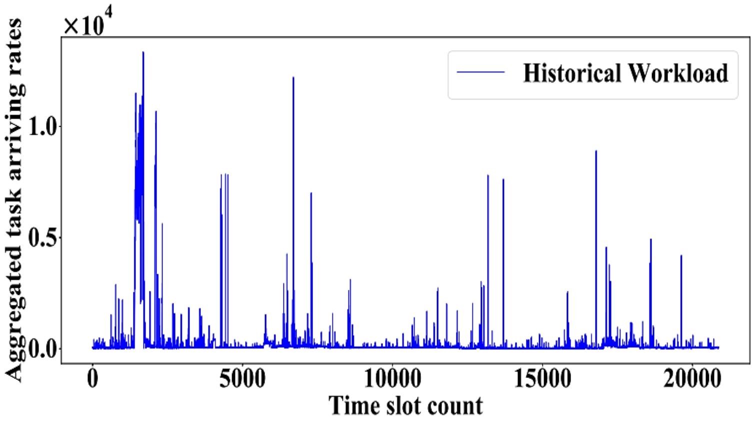

Fig 2: Total Time Series of Workload

Figure 2 is a line graph of the historical workload, over a time. The time slot count, which is between 0 to say about 20,000, is plotted in the x-axis and the rate at which tasks arrive, scaled by a factor of 104 is plotted in the y-axis. The graph demonstrates an extremely dynamic and changing workload, which consists of periodic and acute spikes between intervals of inactivity. There are some major peaks present, the highest being in the area around the 2,500-time slot mark with a rate of more than 12 104. Other significant spikes are observed in and around 5,000, 7,000 and 16,000 time slots. The general trend provides an erratic system load which has specific high demand periods.

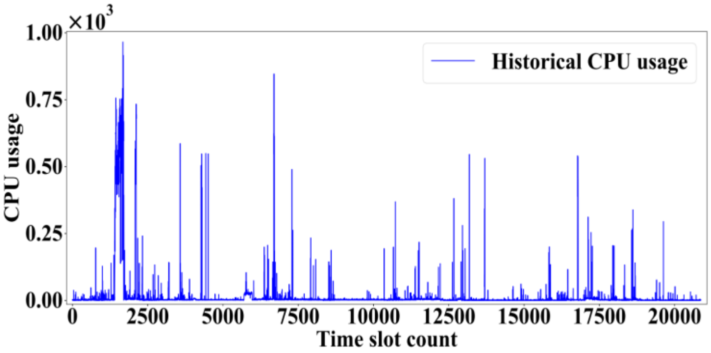

Fig 3: Total Time Series of CPU Usage

Figure 3 is a line chart that displays the past CPU usage. The count of time slots on the x-axis ranges between 0 and 20,000 and the y-axis shows the count of CPU usage multiplied by a value of 103. The graph shows a very active and dynamic trend, with a large number of peaks of CPU utilization, separated by long intervals of extremely low activity. The biggest mountains are related to the 2,000 and 6,500 counts of time slots with the value of approximately 1.00103 and 0.80103 respectively. The general pattern indicates that CPU resources are prone to unpredictable and high demand events which is indicative of intermittent and bursting workload.

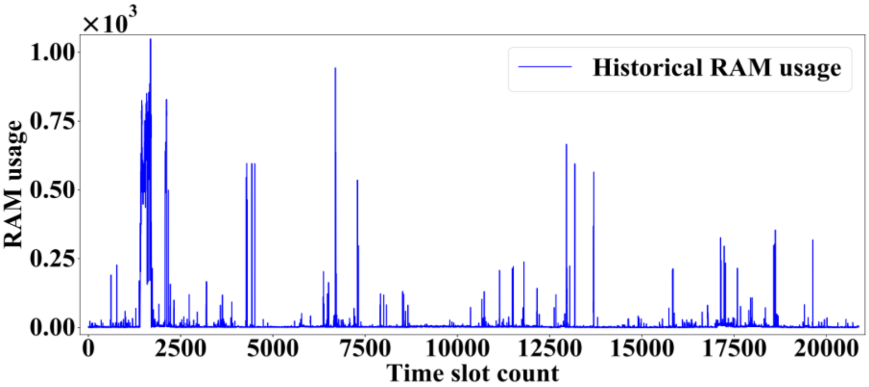

Fig 4: Total Time Series of RAM Usage

Figure 4 is a line graph that illustrates how memory has been utilized over a given period of time. The time slot count, which is between 0 and 20,000, is represented on the x-axis, whereas the RAM usage is represented on the y-axis and is scaled by a factor of 103. According to the graph, the usage pattern is highly dynamic and non-uniform with great spikes that are separated by long intervals of low activity. The presence of prominent peaks can be seen at time slots 2,000 and 7,000 where the usage of RAM is approaching the value of 1.00x103. The data show that the memory requirement of the system is a bursty and sporadic requirement, but not steady and that the periods of high use of the system are not foreseeable.

3.2. Data Preprocessing

Preprocessing of data enhances the quality of raw workload and resource usage data, by removing the noise, outlier, and scaling data. This guarantees the data to be consistent, clean and appropriate to be used to make accurate predictions. The workload and resource usage data used to improve the quality of data and model performance was subjected to the following preprocessing methods:

3.2.1. Logarithmic Scaling

The method of logarithmic data transformation is a method of data transformation, used to normalize the large numeric data to smaller levels, and to stabilize the variance in data sets. The raw data of work load and resource usage is highly deviated and distributed wide [23] and hence modeling is a challenge. In order to normalize the data, each piece of data is subjected to a natural logarithm, causing large values to be compressed and smaller values to be expanded. This change is defined in Equation (1):

$$X' = \ln(X + 1) \tag{1}$$

Where X is the original data and X′ is the transformed value, making the series more uniform.

3.2.2. SG Filter:

The Savitzky-Golay filter cleans up the input signal by eliminating any anomalies or background noise. It raises data accuracy without affecting signal tenor or peak value retention [24]. A time series, according to this theory, ought to be constant with respect to both its mean and variance. There is a need for SG filtering to ensure a stable time series because the resulting time series may contain outliers. Recommend using an SG filter with a window size of 11 for original sequence smoothing because of the positive results it produces. Pictured here is Figure 5:

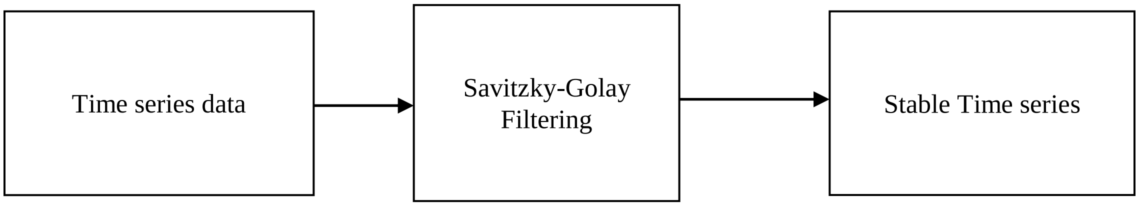

Fig 5: SG Filter

3.2.3. Min-Max Scaling

An essential part of any machine learning preprocessing phase, data scaling changes the range of characteristics so that they all contribute equally to training the model. One common approach to normalization is min-max scaling, which involves transforming data such that it falls inside a predetermined range, usually [0, 1]. Equation (2) shows how it rescales the data by taking the minimum value and dividing it by the whole range:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{2}$$

Where X represents the initial data, X_{min} and X_{max} denote the lowest and highest values in the collection, and X' denotes the output after normalization.

3.3. Data Splitting

The dataset spans 29 days and is split into 60% for training and 40% for testing. This approach ensures the model learns from historical workload and resource usage data.

3.4. Proposed Long Short-Term Memory Model

The LSTM is used to predict time series in long term and it follows the pattern of past entry to predict the workload time series. By adding a memory cell, the LSTM improves the performance of classical RNN method. The input, forget, and output gates make up an LSTM cell [25]. A sigmoid activation function functions as a filter in each gate, deciding what information to retain and what to discard. In addition to the gates, LSTM also has the following components, as shown in Figure 6:

- X_t represents the input data as of time step t.
- H_{t-1} represents the concealed state either from the previous time step or at time step t − 1. In LSTM, it acts as a short-term memory.
- Cell state at time step t−1 is denoted as C_{t-1}. LSTM is controlled by it. Within the LSTM, it undergoes two updates.
- The sigmoid activation function is represented by σ [26].
- The activation function tanh produces results between 1 and -1.
- The forget gate determines whether to remember or forget the previous time step memory C_{t-1}, and its output is f_t, which is a sigmoid function.
- The input sigmoid gate's output determines which incoming data points are added to the cell state.
- The candidate cell state, denoted as $\tilde{c}_t$, is obtained by applying $tanh$ to the external input data X_t, which undergoes a non-linear transformation.

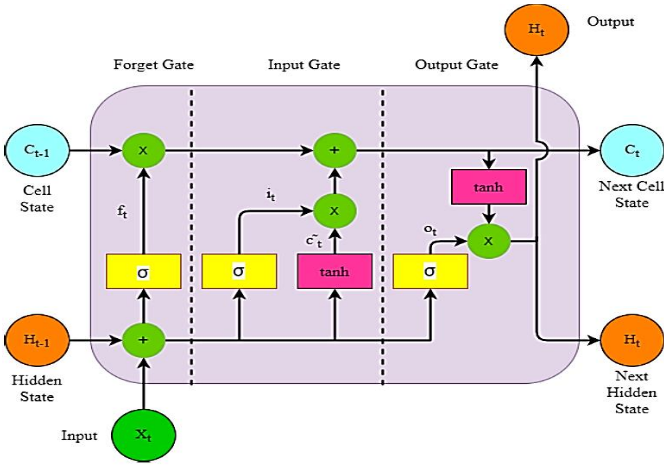

Fig 6: A LSTM Cell Architecture[27]

- The vector o_t is derived from the sigmoid activation of the input vector X_t and the prior hidden state H_{t-1}. Output H_t and concealed state H_t are both affected by this decision, as is the amount of new cell state data.
- C_t represents the subsequent state of the cell.
- H_t serves as the subsequent concealed state and is produced by the LSTM cell.

The following changes are made to the internal gates of an LSTM cell block for every new piece of external input data X_t, as shown in Equations (3) to (6):

$$f_t = \sigma\left(W_f[H_{t-1}, X_t] + b_f\right) \quad (3)$$

$$i_t = \sigma(W_i[H_{t-1}, X_t] + b_i) \quad (4)$$

$$o_t = \sigma(W_o[H_{t-1}, X_t] + b_o) \quad (5)$$

$$\tilde{c}_t = tanh(W_c[H_{t-1}, X_t] + b_c) \quad (6)$$

Finally, an LSTM uses the outcomes of the internal gates and the candidate cell state to update the next cell state C_t and output H_t, as shown in Equations (7) to (8):

$$C_t = f_t \otimes C_{t-1} \oplus i_t \otimes \tilde{c}_t \quad (7)$$

$$H_t = o_t \tanh \otimes (C_t) \quad (8)$$

The LSTM cell effectively combines short-term and long-term memory through its gated architecture, enabling more accurate modeling and prediction of complex time series data compared to traditional models.

3.5. Performance Metrics

The effectiveness of regression models can be measured using different statistical indicators that capture the accuracy and reliability of predictions. The metrics are as follows:

R-Square: The value of the goodness-of-fit between the predicted and true values is represented by R^2 [28]. The accuracy of the predictions is guaranteed when they exactly match the actual data. In this case, $R^2 = 1$. $R^2 \in (-\infty, 1]$ and it is given in Equation (9):

$$R^2 = 1 - \frac{\sum_i(\hat{y}_i - y_i)^2}{\sum_i(y_i - \bar{y}_i)^2} \quad (9)$$

Mean Square Error: The mean squared error (MSE) ranges from 0 to infinity and is calculated as the average of the squares of the differences between the actual and predicted values. MSE = 0 when the expected value is equal to the observed one. Also, the bigger the error involved in prediction, the bigger the MSE. It is computed by Equation (10):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \quad (10)$$

Root Mean Squared Logarithmic Error: RMSLE is appropriate in evaluating the existence of big outliers in the predicted data. RMSLE not as vulnerable to large values in the data as MSE. It falls within [0, +∞] and it is indicated in Equation (11):

$$RMSLE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(y_i + 1)) - (\log(y_i + 1))^2} \quad (11)$$

Overall, all these measures present a balanced perspective of prediction quality, factoring in the general accuracy as well as the influence of the extreme values when comparing to each other.

4. Result Analysis and Discussion

The experimental findings of the Long Short-Term Memory (LSTM) model in proactive resource management of a cloud workload optimization reflected a very high predictive accuracy in all the parameters. With the help of the provided environment Intel Xeon E5-2650 v4 processor with 24 cores, 128 GB RAM, and the NVIDIA Tesla P100 graphics card (16 GB), using Linux Ubuntu 16.04, the frameworks of TensorFlow and Keras. Table 2 shows that for workload, CPU, and RAM predictions, the model got a R² value of 0.99%, which means that the actual and projected values were very close to each other. Workload had an MSE of 13,934.54%, CPU of 122.89%, and RAM of 131.29%, indicating very little variation in prediction errors. Similarly, the Root Mean Squared Logarithmic Error (RMSLE) values remained consistently low (0.15% for workload, 0.16% for CPU, and 0.14% for RAM), highlighting the model's robustness in handling scale-sensitive data. Altogether, the findings verify the effectiveness of the LSTM as an instrument of data-driven resource management in the context of clouds.

Table 2: Performance of LSTM for Cloud Workload Optimization

Long Short-Term Memory Model			
Approaches	R^2	MSE	RMSLE
Workload	0.99	13934.54	0.15
CPU	0.99	128.89	0.16
RAM	0.99	131.29	0.14

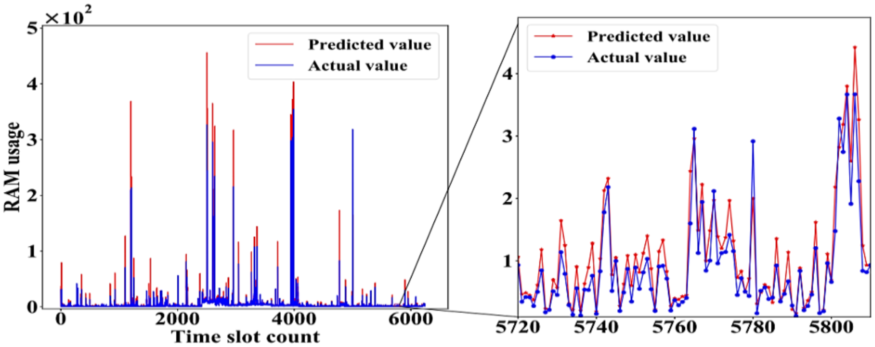

Fig 7: Prediction Result for RAM Usage with LSTM Model

Figure 7 shows a prognostication of an LSTM model on the use of RAM over time slots. The graph where the prediction of the total RAM usage is depicted throughout the period of time approximating about 6000 time slots, the red line is the prediction of the values and the blue line is the actual values. The pattern of the two lines is also similar with some spikes, which means that the model could be used to reveal sudden shifts. The zoom-in view of the right (5720 to 5820) offers a better look and reveals the fact that the predicted values are right on track with real variations. This proves that the LSTM model is useful in predicting dynamic memory usage behavior.

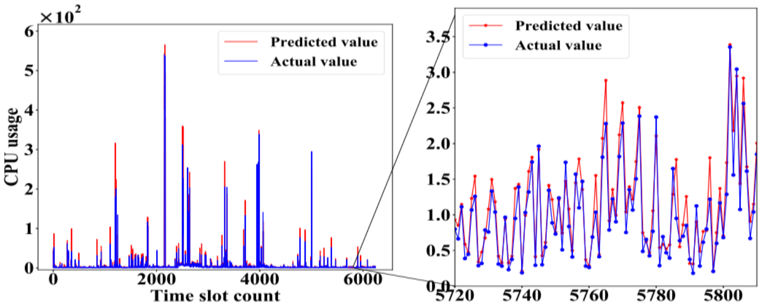

Fig 8: Prediction Result for CPU Usage with LSTM Model

Figure 8 shows a prediction of an LSTM model in terms of CPU usage in time slots. The left graph shows general trends of CPU usage to 6000 time slots where red and blue lines represent the predicted and actual values respectively. These two curves show some similarity in terms of their patterns as they have well-observed peaks indicating the capability of the model to trace workload variations. A closer look is seen in the right-hand zoomed area (time slots 57205820), which points to a point of close comparisons between predicted and actual values; this is able to detect sudden increases and drops. This establishes the efficiency of LSTM model in predicting dynamic trends of CPU utilization.

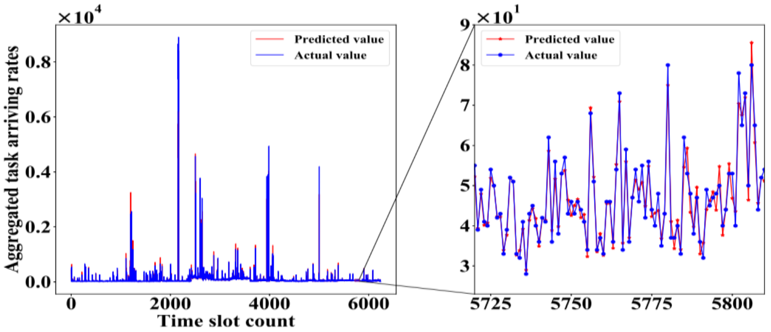

Fig 9: Prediction Result for Workload Time Series with LSTM Model

Figure 9 shows the performance of an LSTM model in predicting workload time series within a number of time slots. The left graph depicts the overall trend of approximately 6000 time slots where the red curve depicts the predicted values and the blue curve depicts actual values. They both have steep peaks and oscillations demonstrating the ability of the model to trace the changes in workloads. The zoomed area on the right (time slots 57255820) has more significant comparison, with the predicted values being largely accurate, hence, being able to reflect dynamic changes and sudden spikes. This proves the accuracy of the LSTM model in modelling and predicting patterns of workload time series.

4.1. Comparative Analysis

The relative study of predictive models of cloud workload optimization demonstrates the high effectiveness of the LSTM fashionable model in contrast to the existing methods. As shown in the Table 3, LSTM achieved an outstanding R² value of 99% across workload, CPU, and RAM predictions, indicating near-perfect accuracy. In contrast, earlier methods such as VAMBig reported lower accuracies of 91.3% for workload, 91.7% for CPU, and 90.9% for RAM, while the SVM model delivered moderate results with 96.83% for workload but considerably lower values of 91.3% for CPU and 86.03% for RAM. The SATCN approach performed better, achieving 98% for workload and 97% for both CPU and RAM, yet it still lagged slightly behind LSTM. These findings demonstrate that LSTM consistently outperforms prior models, establishing it as the most reliable method for proactive resource management in cloud computing environments.

Table 3: Comparative R² Performance of Models for Cloud Workload, CPU, and RAM

	R^2		
Models	**Workload**	**CPU**	**RAM**
VAMBig[29]	91.3	91.7	90.9
SVM[30]	96.83	91.3	86.03
SATCN[31]	98	97	97
LSTM	99	99	99

The experimental results show that the LSTM model is useful in cloud contexts for predicting workload, CPU, and RAM utilization. Its predictive strength surpasses conventional models, confirming the value of temporal dependency modeling for resource management. The results emphasize that proactive strategies not only enhance SLA compliance but also contribute to sustainable energy use. However, extending the framework to heterogeneous, multi-cloud settings remains a challenge, as real-world cloud workloads exhibit diverse characteristics.

5. Conclusion & Futurework

Accurate modeling of cloud workload behavior is central to reducing energy consumption, maintaining SLA compliance, and ensuring efficient use of resources. The proposed LSTM-based framework achieves superior predictive performance, delivering an R² of 0.99 for workload, CPU, and RAM, alongside low MSE values of 13,934.54, 128.89, and 131.29, and RMSLE scores of 0.15, 0.16, and 0.14, respectively. These results highlight the framework's capability to minimize error, prevent over-provisioning, and proactively optimize resource allocation. A comparative analysis underscores the LSTM's advantage over prior models, including VAMBig, SVM, and SATCN. While VAMBig achieved accuracies around 91%, SVM performed moderately with CPU and RAM utilization, and SATCN achieved 97–98%, the LSTM consistently outperformed all, establishing itself as the most reliable method for workload optimization. In enterprise-scale deployments, it stands out due to its adaptability to changing workloads and its capacity to record complicated temporal connections. Future work should address extending this approach to heterogeneous multi-cloud and federated environments, where workload diversity is more challenging. Incorporating hybrid architectures, reinforcement learning, and evolutionary algorithms could further strengthen adaptability. Real-time deployment integrated with automated migration and autoscaling strategies presents a promising direction for creating fully autonomous, intelligent cloud workload optimization systems.

Reference

[1] M. G. Avram, "Advantages and Challenges of Adopting Cloud Computing from an Enterprise Perspective," *Procedia Technol.*, 2014, doi: 10.1016/j.protcy.2013.12.525.

[2] V. Prajapati, "Cloud-Based Database Management:

Architecture, Security, challenges and solutions," *J. Glob. Res. Electron. Commun.*, vol. 01, no. 1, pp. 07–13, 2025.

[3] V. Varma, "Secure Cloud Computing with Machine Learning and Data Analytics for Business Optimization," *ESP J. Eng. Technol. Adv.*, vol. 4, no. 3, 2024, doi: 10.56472/25832646/JETA-V4I3P119.

[4] M. Pantazoglou, G. Tzortzakis, and A. Delis, "Decentralized and energy-efficient workload management in enterprise clouds," *IEEE Trans. cloud Comput.*, vol. 4, no. 2, pp. 196–209, 2015.

[5] V. Singh, "Reinventing Business with Cloud Integration: The Cost - Effectiveness of Replacing Legacy Applications," *Int. J. Sci. Res.*, vol. 13, no. 8, pp. 1882–1887, 2024.

[6] K. Anderson, "Multi-Agent Reinforcement Learning for Enterprise Cloud Workload Optimization," 2023.

[7] G. Maddali, "An Efficient Bio-Inspired Optimization Framework for Scalable Task Scheduling in Cloud Computing Environments," *Int. J. Curr. Eng. Technol.*, vol. 15, no. 3, pp. 229–238, 2025.

[8] S. Narang and V. G. Kolla, "Next-Generation Cloud Security: A Review of the Constraints and Strategies in Serverless Computing," *Int. J. Res. Anal. Rev.*, vol. 12, no. 3, pp. 1–7, 2025, doi: 10.56975/ijrar.v12i3.319048.

[9] V. M. L. G. Nerella, "A Database-Centric CSPM Framework for Securing Mission-Critical Cloud Workloads," *Int. J. Intell. Syst. Appl. Eng.*, vol. 10, no. 1, pp. 209–217, 2022.

[10] A. R. Duggasani, "Scalable and Optimized Load Balancing in Cloud Systems: Intelligent Nature-Inspired Evolutionary Approach," *Int. J. Innov. Sci. Res. Technol.*, vol. 10, no. 5, pp. 2153–2160, May 2025, doi: 10.38124/ijisrt/25may1290.

[11] R. Dattangire, R. Vaidya, D. Biradar, and A. Joon, "Exploring the Tangible Impact of Artificial Intelligence and Machine Learning: Bridging the Gap between Hype and Reality," in *2024 1st International Conference on Advanced Computing and Emerging Technologies (ACET)*, IEEE, Aug. 2024, pp. 1–6. doi: 10.1109/ACET61898.2024.10730334.

[12] A. Birhade, V. Shejul, D. Chavan, N. Y. Patil, and D. R. D. Jadhav, "AI and Machine Learning in Cloud Optimization." 2025. doi: 10.2139/ssrn.5321423.

[13] R. Q. Majumder, "Machine Learning for Predictive Analytics: Trends and Future Directions," *Int. J. Innov. Sci. Res. Technol.*, vol. 10, no. 4, pp. 3557–3564, 2025, doi: 10.38124/ijisrt/25apr1899.

[14] N. Prajapati, "The Role of Machine Learning in Big Data Analytics: Tools, Techniques, and Applications," *ESP J. Eng. Technol. Adv.*, vol. 5, no. 2, pp. 16–22, 2025, doi: 10.56472/25832646/JETA-V5I2P103.

[15] G. Modalavalasa and H. Kali, "Exploring Big Data Role in Modern Business Strategies: A Survey with Techniques and Tools," *Int. J. Adv. Res. Sci. Commun. Technol.*, vol. 3, no. 3, pp. 431–441, Jan. 2023, doi: 10.48175/IJARSCT-11900B.

[16] A. Sharma and S. Kabade, "AI-Driven and Cloud-Enabled System for Automated Reconciliation and Regulatory Compliance in Pension Fund Management," *Int. J. All Res. Educ. Sci. Methods*, vol. 12, no. 6, pp. 3019–3027, 2024.

[17] Q. Xin, "A Deep Reinforcement Learning Approach to Optimizing Cloud Workload Migration," *Am. J. Interdiscip. Res. Innov.*, vol. 4, no. 3, pp. 10–15, 2025, doi: 10.54536/ajiri.v4i3.5429.

[18] R. Karthikeyan, S. R. A. Samad, V. Balamurugan, S. Balasubaramanian, and R. Cyriac, "Workload Prediction in Cloud Data Centers Using Complex-Valued Spatio-Temporal Graph Convolutional Neural Network Optimized With Gazelle Optimization Algorithm," *Trans. Emerg. Telecommun. Technol.*, vol. 36, no. 3, Mar. 2025, doi: 10.1002/ett.70078.

[19] C. Diwaker and N. Miglani, "Optimizing Autoencoder for Workload Prediction in Cloud Environment Using Particle Swarm Optimization," in *The International Conference on Recent Innovations in Computing*, 2024, pp. 185–205.

[20] T. Ali, H. U. Khan, F. K. Alarfaj, and M. Alreshoodi, "Hybrid deep learning and evolutionary algorithms for accurate cloud workload prediction," *Computing*, vol. 106, no. 12, pp. 3905–3944, 2024.

[21] Z. Ahamed, M. Khemakhem, F. Eassa, F. Alsolami, A. Basuhail, and K. Jambi, "Deep Reinforcement Learning for Workload Prediction in Federated Cloud Environments.," *Sensors (Basel).*, vol. 23, no. 15, Aug. 2023, doi: 10.3390/s23156911.

[22] J. Dogani, F. Khunjush, M. R. Mahmoudi, and M. Seydali, "Multivariate workload and resource prediction in cloud computing using CNN and GRU by attention mechanism," *J. Supercomput.*, vol. 79, no. 3, pp. 3437–3470, 2023.

[23] S. Priyadarshini, T. N. Sawant, G. Bhimrao Yadav, J. Premalatha, and S. R. Pawar, "Enhancing security and scalability by AI/ML workload optimization in the cloud," *Cluster Comput.*, vol. 27, no. 10, pp. 13455–13469, Dec. 2024, doi: 10.1007/s10586-024-04641-x.

[24] P. Rawat, "Workload prediction for cloud services by using a hybrid neural network model," National College of Ireland, 2022.

[25] S. Simaiya *et al.*, "A hybrid cloud load balancing and host utilization prediction method using deep learning and optimization techniques," *Sci. Rep.*, vol. 14, no. 1, p. 1337, 2024.

[26] Y. Xing, "Work scheduling in cloud network based on deep Q-LSTM models for efficient resource utilization," *J. Grid Comput.*, vol. 22, no. 1, p. 36, 2024.

[27] M. E. Karim, M. M. S. Maswood, S. Das, and A. G. Alharbi, "BHyPreC: A Novel Bi-LSTM Based Hybrid Recurrent Neural Network Model to Predict the CPU Workload of Cloud Virtual Machine," *IEEE Access*, 2021, doi: 10.1109/ACCESS.2021.3113714.

[28] A. Lopez Garcia *et al.*, "A Cloud-Based Framework for Machine Learning Workloads and Applications," *IEEE Access*, vol. 8, pp. 18681–18692, 2020, doi:

10.1109/ACCESS.2020.2964386.

[29] J. Bi, H. Ma, H. Yuan, and J. Zhang, "Accurate Prediction of Workloads and Resources with Multi-Head Attention and Hybrid LSTM for Cloud Data Centers," *IEEE Trans. Sustain. Comput.*, vol. 8, no. 3, pp. 375–384, 2023, doi: 10.1109/TSUSC.2023.3259522.

[30] Z. Amekraz and M. Y. Hadi, "CANFIS: A Chaos Adaptive Neural Fuzzy Inference System for Workload Prediction in the Cloud," *IEEE Access*, vol. 10, pp. 49808–49828, 2022, doi: 10.1109/ACCESS.2022.3174061.

[31] H. Yuan, S. Member, J. Bi, S. Member, S. Li, and S. Member, "An Improved LSTM-Based Prediction Approach for Resources and Workload in Large-Scale Data Centers," vol. 11, no. 12, pp. 22816–22829, 2024.