

ABHISHEK KUMAR PANDEY
SARVPAL SINGH

AN EFFICIENT WORKLOAD PREDICTION WITH MINIMIZED SLA VIOLATIONS IN CLOUD COMPUTING USING DEEP LEARNING MODEL

Abstract *Accurate workload forecasting is crucial for optimizing resource distribution and minimizing cloud Service Level Agreement (SLA) breaches. Owing to their sequential and linear nature, existing models, such as Long Short-Term Memory, CNN, and their variants fail to learn complex temporal patterns and relationships, resulting in inferior prediction accuracy and SLA violations. This study presents a novel BiLSTM-based multi-modal attention model (BiLSTM-MA) to address these challenges. The hybrid model integrates Bidirectional Long Short-Term Memory networks with a multi-modal attention mechanism to enhance the feature extraction from diverse cloud workload data. Comprehensive trials demonstrated that BiLSTM-MA markedly outperformed leading models, including Deep Long Short-Term Memory, CNN-LSTM, MLSTM, and LSTM, across several parameters such as SLA violations, energy consumption, average CPU utilization, RMSE, and RMSPE. The results indicate that the proposed model outperforms all baseline models across all workloads. The proposed model exhibits the lowest average CPU use, incurs the fewest SLA violations, and demonstrates minimal energy consumption, signifying its superior resource management capabilities while properly forecasting demand. The mean CPU usage for BiLSTM-MA within the workload was 64%, indicating a reduction in RMSE error rates of 8.14%, 46%, 48%, and 55.39% compared to baseline approaches.*

Keywords Deep Learning Technique, SLA Violation, BiLSTM, Cloud Computing, Workload prediction

Citation Computer Science 27(2) 2026: 27–46

Copyright © 2026 Author(s). This is an open access publication, which can be used, distributed and reproduced in any medium according to the Creative Commons CC-BY 4.0 License.

1. Introduction

Cloud computing has changed the fundamentals of business and individuals' consumption and management of computational resources. The challenge is to use resources efficiently without the ability to maintain SLAs violation. Workload prediction models can predict resource requirements promptly so that the efficient and elastic allocation of cloud resources becomes possible. Unlike cloud workloads exhibiting strong signal non-linearity, traditional approaches/models, and machine learning methods, workload prediction delivers limited performance owing to oversimplistic assumptions. Consequently, there is an increasing demand to investigate learning methods that may have the potential for better prediction performance through more efficient modeling of complex temporal and spatial relationships [14]. These challenges can be solved by workload prediction, which uses historical workload patterns to predict the demand for resources more accurately. Predicting the workload behavior helps to provide and allocate resources beforehand, ensuring that enough is available to meet the expected demand but not over-provisioned to keep unnecessary costs [5]. However the approach must enhance its predictive accuracy in a significantly dispersed cloud environment. In such environments, a prediction model provides interval values for future energy usage, CPU utilization, and SLA violations, encompassing both near-and long-term scenarios. In a cloud environment, each job is assigned to a different machine, and customers are responsible for supplying CPU, energy, and I/O resources. This arrangement ensures project success but poses challenges for projecting the data center CPU and energy consumption because customers may only sometimes know their exact resource requirements. The primary criterion for accuracy in this context is the CPU and energy usage of the cloud service provider, as it directly affects work and cloud performance [15]. Proactive resource management relies on a cloud environment to distribute resources dynamically based on demand forecasts. Accurate capacity planning of workloads is paramount for supporting cloud providers to mitigate SLA violations, optimize resource utilization, and reduce energy consumption [19]. Models such as Autoregressive Integrated Moving Average (ARIMA), Exponential Smoothing, and other traditional time-series prediction models have limitations in predicting the non-linearity of cloud workloads. Models based on deep learning, such as long short-term memory (LSTM), CNN-LSTM, Deep LSTM (DLSTM), and Modified LSTM (mLSTM), have proven to be effective for workload prediction because of their capability to capture long-term dependencies. Nevertheless, such models lack support for multi-modal data and perform less effectively under rapid workload changes, resulting in inferior performance [11]. This paper presents a novel multi-modal self-attention model based on BiLSTM for cloud workload prediction. BiLSTM-MA uses a multi-modal self-attention-based attention mechanism that enables the model to concentrate on the most critical features obtained from various types of information sources while utilizing the power of Bidirectional LSTM (BiLSTM) networks to handle sequential dependencies in both forward and backward directions. Given that our prediction problem involves handling heterogeneous data,

including network statistics, CPU, and memory consumption, the multi-modal self-attention component can assist the model in learning more efficient representations, leading to improved prediction performance. The publicly accessible Google Cloud Trace 2019 dataset, which monitors resource usage from a Google Cloud cluster, was used to test the the proposed BiLSTM-MA model. This dataset is appropriate for assessing workload prediction models because it includes comprehensive statistics on CPU utilization, memory usage, disk read/write operations, and network bandwidth. The tests evaluated the performance of BiLSTM-MA against cutting-edge techniques such as CNN-LSTM, mLSTM, LSTM, and DLSTM-MA in terms of important metrics such as average CPU usage, energy consumption, SLA violation rates, Root Mean Square Error(RMSE) and Root Mean Square Percentage Error(RMSPE). The significant contributions of this research are highlighted as follows:

- Deep Learning based multi-modal attention model developed for cloud workload prediction.
- BiLSTM-MA outperforms state-of-the-art techniques like CNN-LSTM, mLSTM, LSTM, and DLSTM-MA in various parameters, including energy usage, SLA violations, and average CPU utilization.
- Experiments with various workloads demonstrated the capability of the model to diminish SLA breaches, save energy costs, and optimize resource management.
- The BiLSTM-MA model effectively addresses workload management in dynamic cloud infrastructures, improves service quality, reduces costs, and promotes sustainability.

This paper is structured as follows: Section 2 examines the studies. Section 3 describes the architecture and design. Section 4 presents observations and outcomes. Finally, Section 5 outlines the conclusions and the subsequent steps.

2. Related work

Recent research has examined workload and forecasting prediction in cloud computing. This section outlines current research on this subject, covering both conventional methods and the development of deep learning strategies. The literature has focused on workload prediction in cloud systems, highlighting its function in maintaining service standards, improving the effectiveness of resource distribution and exploring various techniques, including state-of-the-art deep-learning technologies [8] [9] [1].

Al-Sayed et al. [1] proposed a deep learning approach that, despite its limitations, has the potential to provide more reliable results for long-term forecasting. Conversely, the Sequence-to-Sequence neural machine translation approach, sometimes called Seq2Seq, effectively understands long text. This study treats the estimation of activity succession as a translation issue. Therefore, the proposed approach based on Attention Seq2Seq is recommended for predicting workloads related to cloud resources. This study used a dataset from a Google Network of 11,000 machines to verify the proposed technique. A unique approach called cumulative validation was

proposed to improve the suggested framework, making it more effective than cross-validation.

Joeng et al. [7] forecast memory usage and CPU utilization using a Bi-LSTM model with an attention mechanism. These predictions were based on whether the patterns were similar. ReVPA, vertical pod autoscaling, or reallocation of container resources adjusts resources based on predictions before consumption. Lastly, HPA to avoid overflow works incredibly fast, taking pod information and predicting resource usage into account and balancing the resources promptly. While looking into how ProHPA works, it also turned out that the average CPU and RAM used by ProHPA had increased to 23.39% and 42.52% respectively.

Xu et al. [23] proposed an efficient technique for workload projection called Deep Neural Network (esDNN). The sliding window transforms multidimensional data into a structured learning time series. The esDNN leads to data being at a suitable level for deep learning algorithms to analyze. The initial tracks were sourced from cloud data centers stored by Alibaba and Google, and extensive testing was conducted based on these findings. These tests were designed to demonstrate the effectiveness of esDNN. The experimental results show that the esDNN can be used for accurate and fast workload predictions in cloud environments with a high degree of reliability. The esDNN exhibited a mean square error reduction (up to 15% against the latest baselines). This is different from the method that uses only GRU.

Yazdanian et al. [24] developed a hybrid model using a Generative Adversarial Network (GAN) to predict a highly volatile and chaotic cloud workload. This methodology employs a multistep-ahead approach, considering both the composite and nonlinear dependencies between future samples of the workload time series. Consequently, it generates considerable gains in workload prediction accuracy which may be useful for informing resource management decisions. E2LG consists of two major components: an LSTM stacked approach and the forecasting style of the IMF units. However, high-frequency IMF units can be predicted using the GAN design.

Kumar et al. [10] introduced a hybrid strategy that combined LSTM and RNN deep-learning methods. It has been used to address problems associated with the dynamic scalability of resources and power consumption in CDCs. The RNN contains networks with connective loops that allow information to circulate and remain in the form of memories. In the loop, each network receives data and inputs from the previous network, performs a task and generates an output. It also passes the data to the following network within the cycle. Some applications may require only current information, whereas others expect additional historical knowledge. For example, learning is delayed for recurrent neural networks (RNN) because the previous information it needs to access may be astronomically distant when it requires such data. Hence, to eliminate this issue, the proposed method implements LSTM networks. This is meant to avoid the long-term dependence problem observed in recurrent networks. An LSTM network can store information for a long time.

Simaiya et al. [22] developed an advanced prediction model based on a neural

network to achieve a better accuracy in scaling operations. The proposed method first classified the VMs based on their historical use features and forecasted their future use patterns for future time intervals. To achieve this, the program employed a multi-layered perceptron classification technique. Sharma et al. [21] proposed performance aware automatic scaling that can be used for the elastic architecture of a cloud based on containerized cloud-based operations. This enables dynamic allocation of available resources based on demand variability.

Lilhore et al. [12] developed a generalized scaling model of memory and CPU estimates for VM resource utilization using CNN and K-means. This method precisely predicts future workload requirements. Using a Bayesian method, the investigation predicted resource consumption for a virtual machine, typically throughout a workday. This recurrent LSTM neural network model is a predictive approach for estimating future HTTP workloads. The proposed method is an artificial neural network (ANN), that prevents the scaling challenge posed by elastic performance and autonomous applications.

Sharma et al. [21]] used a multi-objective adaptive algorithm to effectively predict the memory usage, CPU usage, and energy consumption within the following period. Ghasemi et al. [6] created a Long Short-Term Memory (LSTM) technique to predict the workload of cloud datacenters.

Pawar et al. [18] addressed the problem of load prediction in cloud datacenters. An LSTM network was used to develop a computational load-forecasting model. According to the experimental results, the recommended strategy demonstrated more accuracy in prediction than the other alternatives that were also evaluated. Furthermore, several cloud service providers use user-defined resource criteria to provide auto-scaling capabilities, restricting their capacity to construct models based on diverse workload characteristics.

This literature survey showed that deep learning and classical methods have been widely applied to workload and forecasting predictions in cloud environments. Deep-learning techniques, specifically DCNN and DLSTM, have shown promise. These deep learning models demonstrate their ability to understand complex patterns, adapt to workload fluctuations, and improve the prediction accuracy. These models require additional testing to fully explore their potential and to develop hybrid approaches that blend deep and classical learning [2, 3, 25]. The pertinent literature, machine learning, and deep learning resource predictions are listed in Table 1.

Table 1
An overview study of workload prediction on deep learning

Study Ref.	Techniques	Performance evaluation	Objective	Research gap
Devi et al. [3]	ANN and ARIMA Techniques	Memory usage, CPU utilization, RMSE, and MAPE	Examine the prediction load and the implementation of nonlinear and linear prediction techniques inside cloud computing settings	There is a research gap in assessing several necessary measures, including determining SLA violations
Liu et al. [13]	ARIMA and LSTM algorithm	CPU Usage, Optimal forecast	Resource conservation and workload prediction	Memory utilization in hybrid models with nonlinear components is subject to limitations
Ding et al. [4]	RACC-MDT Technique	CPU and memory usage	Enhanced memory utilization and energy usage	Overhead, migration delays, and SLA violations are disregarded
Sagheer et al. [20]	Deep LSTM	Prediction accuracy and MAPE	This technique guarantees enhanced learning results by integrating complex conditional distributions for time series data	Time series data that is reliant on several variables is not evaluated
Pattanayak et al. [17]	Fuzzy and SVM algorithm	Accuracy prediction for Friedman test, rank test, and Nemeny test	This work presents a data-driven technique for predicting time series by fuzzy membership values	The accuracy of the forecasting model might be enhanced further

Table 1 (cont.)

Study Ref.	Techniques	Performance evaluation	Objective	Research gap
Ouhame et al. [16]	CNN-LSTM	Usage of memory, CPU, and network performance	Enhancing resource efficiency while concurrently reducing the need for relocations	There have been discrepancies in load forecast, which necessitated changing settings

3. Materials and methods

3.1. Data

A dataset from Google's 2019 data trial for one of its cloud clusters was used to evaluate the BiLSTM-MA model. This dataset includes performance parameters, including CPU and memory consumption, disk operation network bandwidth, and job arrival and completion times for the clusters' working units over 24 h. The BiLSTM-MA model's evaluation dataset was taken from the 2019 data trial of a Google Cloud cluster. It offers information on resource usage, such as CPU, RAM, disk I/O, and network packet usage. It provides high-fidelity cloud resource usage, making it ideal for testing workload-prediction methods in dynamic and realistic cloud environments.

3.2. Data pre-processing

Data preparation is necessary so that the data gathered from Google's data tracking system can be appropriately implemented in the BiLSTM-MA model. This procedure entails data cleansing by identifying duplicates or entries and filling in any gaps or null values to guarantee a correct dataset. The next step is normalization, which often scales the metric values (such as CPU, memory, and I/O) within a given range $[0 - 1]$.

This guarantees the rapid convergence of the learning models. These data were then divided using a sliding window technique based on predetermined time frames to capture the correlations. Every window consists of observations made during a given period with a particular overlap. Feature engineering improves the model's performance by producing metrics (such as averages and maximum values), developing new metrics (such as CPU/memory ratio), and adding time-related features. These preprocessing steps transform the data into a consistent format in which the DCNN DLSTM model can be used for efficient training and forecasting.

3.3. Proposed BiLSTM-MA model design and architecture

The approach uses the BiLSTM-MA model to learn discriminative features and capture intricate temporal correlations from diverse workload data of cloud environments. To maximize workload prediction while minimizing SLA set violations, this innovative

architecture combines the advantages of both BiLSTM networks and a multi-modal attention mechanism. Fig. 1 depicts the operation of the proposed BiLSTM-MA model design architecture. The input data to the model are multimodal and include the most interesting workload metrics, such as CPU usage, memory usage, network traffic, and disk I/O stats. These heterogeneous data sources must be preprocessed and maintained on a homogenous scale. The model's heart is a bidirectional network with two LSTM layers: forward and backward. The BiLSTM layer captures forward and backward dependencies, allowing for a greater context of temporal characteristics in the workload data. This will also significantly improve the model's recognition of complex trends and trend reversal.

Let $\vec{h}_t$ be the hidden state of the forward LSTM cell at each time step t , and $\overleftarrow{h}_t$ be the hidden state of the backward LSTM cell. The overall output at time step t is equal to the concatenation of the previous and current hidden states, as expressed in Equation (1).

$$h_t = [\vec{h}_t ; \overleftarrow{h}_t] \quad (1)$$

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (3)$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (4)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (5)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (6)$$

$$h_t = o_t * \tanh(C_t) \quad (7)$$

In the above Equations (2)–(7), f_t is the forget gate, which determines which information from the previous cell state C_{t-1} is discarded. i_t represents the input gate that determines the new information added to the cell state. $\tilde{C}_t$ represents the candidate state to add potential new information in the cell, and the updated cell state is denoted by C_t , which combines the decision of the forget gate and the candidate cell state. Controlling the information using the output gate is represented as O_t , which passes the information to the following and final hidden states. h_t at time step t is passed on to the next layer. The weight matrices for the respective gates are denoted as W_f , W_i , W_C , and W_o . The respective biases were b_f , b_i , b_C , and b_o .

3.3.1. Multi-modal attention model

After extracting the output of the BiLSTM model, a multimodal self-attention mechanism was used to enhance prediction accuracy. It comprises three distinct pooling mechanisms: Max Pooling, Average Pooling, and Sum Pooling, succeeded by softmax. Incorporating these pooling procedures aims to obtain different insights from

the BiLSTM outputs, thereby facilitating the input necessary for making more comprehensive decisions. Upon completion of the output layer's processing via BiLSTM, extracting the most salient features is essential, accomplished by a technique known as Max Pooling, which involves selecting the maximum value from each feature map. This highlights the most influential aspects that are critical for forecasts. Conversely, Average Pooling computes the mean of feature values, yielding a more comprehensive representation that encapsulates the predominant trend in the data. We averaged them by matching them in the same sequence, reducing the noise impact and providing an equitable representation of the characteristics. Conversely, Sum Pooling retains the aggregate contribution of all attributes at a given time step by summing their values. This is particularly advantageous when the ultimate forecast relies on the aggregate impact of characteristics. The MA model employs three pooling procedures, enabling it to collect the entire contribution of the features, identify the most significant factors, and discern the general trend of the combined information.

This helps a model to focus on critical interactions that occur over time. These scores are called the attention weights.

$$w_t = \frac{\exp(a_{\text{score}})}{\sum_{t'} \exp(a_{\text{score}}^{t'})} \quad (8)$$

In Equation (8), w_t represents the attention weight for time step t , and a_{score} is the alignment score for time step t , which is evaluated by the forward pass of input as h_t . We used max and average pooling over the attention scores to fine-grain the attention mechanism. Following pooling, these results are concatenated together by the softmax activation function, ensuring that it provides a probability of attention to an input weight with a total score of one;

$$w_{\max} = \max_t w_t \quad (9)$$

$$w_{\text{avg}} = \frac{1}{T} \sum_{t=1}^T w_t \quad (10)$$

$$w_{\text{sum}} = \sum_{t=1}^T w_t \quad (11)$$

$$w_o = \text{softmax}([w_{\max}, w_{\text{avg}}, w_{\text{sum}}]) \quad (12)$$

In the above Equations (9)–(11), the maximum attention score is denoted by $w_{\max}$ the average attention score is denoted by w_{avg} and the total number of time steps in the sequence is denoted by T from max pooling and average pooling. w_o represents the pooled value combinations using the SoftMax function.

At each time step, the attention score of high-level features was computed as a pooled representation of all potentials, allowing the market stock price data to

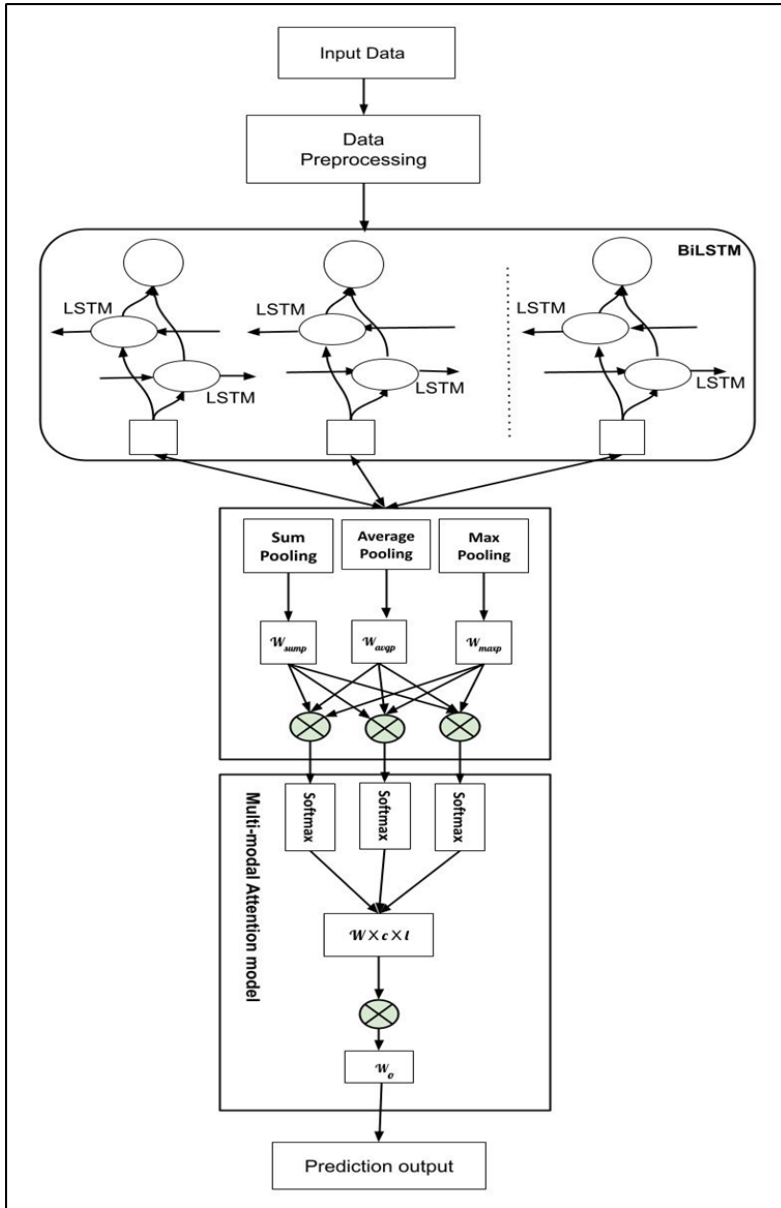


Figure 1. Working process of the proposed method design architecture

emphasize either global or local features. Finally, the output of the multi-mode self-attention mechanism is produced as a weighted sum of the hidden states from the BiLSTM layer, incorporating both time and context attention.

$$\hat{h}_c = \sum_{t'} w_o h_{t'} \quad (13)$$

Here, $\hat{h}_t$ denotes the weighted sum of the attention output and the BiLSTM layer output, incorporating the final outcome of the proposed model. The output layer predicts the workload for the next time step(s). It estimates essential metrics such as CPU and memory utilization, which support proactive resource allocation decisions while minimizing the risk of SLA violations.

4. Experimental setup and results discussion

The experimental configuration was based on the Google Cloud Trace 2019 dataset [19], which contains resource utilization records from a real-world cloud cluster. The comparison was carried out at both practical and synthetic levels using various data models, including the proposed BiLSTM-MA. The dataset includes dynamic workload metrics such as CPU utilization, memory usage, disk I/O, and network traffic, making it suitable for evaluating models designed for cloud workload prediction.

Before training, the dataset was preprocessed by filling missing values using linear interpolation and applying min-max normalization to scale all features between 0 and 1. To capture temporal dependencies effectively, the dataset was split using a sliding window approach, where a sequence of t time steps was used as the input to predict the next time step.

The proposed BiLSTM-MA architecture was implemented using a Python-based deep learning library. Empirical testing and optimization were conducted to determine the key parameters: the BiLSTM layer used 128 units, dropout was set to 0.2 to reduce overfitting, and four attention heads were employed in the multi-modal attention module to allow the model to focus on different aspects of the input features simultaneously. Table 2 lists the workloads and parameter settings used in this experiment.

The BiLSTM-MA model was evaluated using four different workloads: Workload 1, comprising 50 virtual machines and 50 provisioning managers; Workload 2, consisting of 100 virtual machines and 100 provisioning managers; Workload 3, comprising 150 virtual machines and 150 provisioning managers; and Workload 4, comprising 200 virtual machines and 200 provisioning managers. The primary focus of the evaluation was based on CPU usage, energy consumption, and memory utilization. The performance of the BiLSTM-MA model was compared with that of four state-of-the-art models: DLSTM, CNN-LSTM, mLSTM, and LSTM.

4.1. Evaluation metrics

The performance of the BiLSTM-MA model was evaluated by comparing it with baseline models such as DLSTM, CNN-LSTM, mLSTM, and LSTM using a test

Table 2
Setup of the simulation parameters

BiLSTM	128 LSTM units for each direction
Learning rate	0.001
Pooling strategy	Max pooling, Average pooling, Sum pooling
Dropout rate	0.3
Activation function	ReLU
Loss function	MSE
Attention head	4
Batch size	64
Epoch	100
Workload	4

dataset. Evaluation metrics such as Energy Consumption, SLA Violation Rates, Average CPU Utilization, RMSE, and RMSPE were used to assess its effectiveness.

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N |T_e(y_i) - O_p(y_i)|^2} \quad (14)$$

$$RMSPE = \sqrt{\frac{1}{N} \sum_{i=1}^N \left(\frac{T_e(y_i) - O_p(y_i)}{T_e(y_i)} \right)^2} \quad (15)$$

where $T_v(y_i)$ is the targeted real value and $O_p(y_i)$ is the predicted value. The total number of samples is denoted by N in Eqs. 14 and 15.

Energy Consumption: BiLSTM-MA used lower energy than the other models for all workloads, as shown in Fig. 2, which is essential for minimizing operational costs in cloud data centers. BiLSTM-MA consumed 1000 Watts for Workload 1, compared to other models, with values ranging from 2083 Watts (LSTM) to 4056 Watts (mLSTM). For Workload 2, BiLSTM-MA consumed 1559 Watts whereas DLSTM consumed 4403 Watts and LSTM consumed 4635 Watts, which is a huge difference. Similarly, the Introduction of each Workload 3 BiLSTM-MA to this workload saved 4245 Watts vs BiLSTM-MA at Watt level 2674 (mLSTM = 4226 Watts). For Workload 4, mLSTM consumed 3723 Watts and LSTM consumed 4100 Watts, with BiLSTM-MA using only 1293 Watts, again significantly lower than those of the other models. This shows that BiLSTM-MA performs well in saving energy, which benefits from cost reduction and environmental protection.

Average CPU Utilization: As seen from the results in Fig. 3, BiLSTM-MA has consistently lower average CPU utilization across all workloads than the other models. The average CPU utilization for BiLSTM-MA at Workload 1 was 64%. In comparison, it is 77% and higher than that of the other models, demonstrating high resource

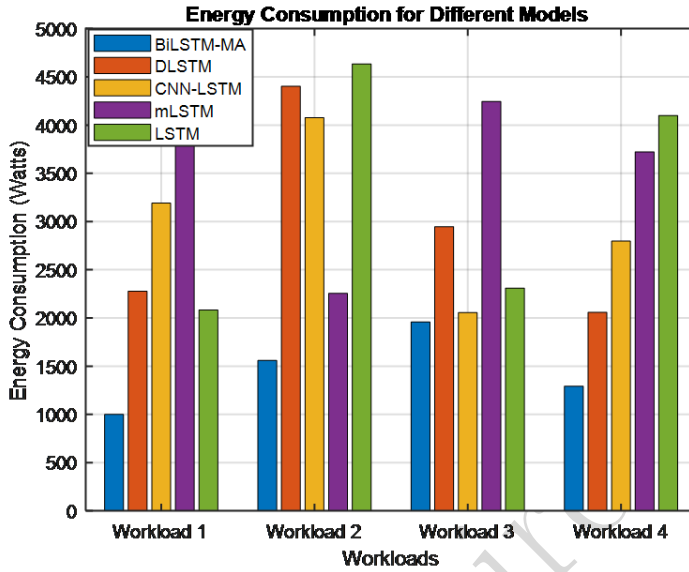


Figure 2. Energy consumption analysis of proposed method and other methods

management ability by consuming a low CPU percentage. For Workload 2, BiLSTM-MA can keep the CPU utilization no higher than 67%, whereas the other two models (CNN-LSTM and mLSTM) exceeded the 88% point of inefficiency. Similar trends can be observed in Workloads 3 and 4, where BiLSTM-MA showed a relatively lower CPU utilization of 7852% for both workloads. This also translates into better efficiency, as lower CPU utilization means that the model spreads the load across virtual machines and does not cause spikes in resource consumption.

SLA Violation: The BiLSTM-MA model has the least SLA violations in all four workloads as shown in Fig. 4, proving its capability to predict and allocate resources to satisfy the workload demand. BiLSTM-MA, on the other hand, had only four SLA violations in Workload 1, and LSTM had 19. In Workload 2, the corresponding number of SLA violations was even lower for BiLSTM-MA: only two vs. CNN-LSTM at 10 and mLSTM at 19, therefore the BiLSTMs present a significant advantage over these models. The trend persisted in Workload 3, for which BiLSTM-MA achieved the lowest SLA violations (2), again outperforming DLSTM, CNN-LSTM, mLSTM and LSTM on this metric with more significant margins of performance (which yielded significantly higher numbers of SLA violations: 16 for DLSTM; 18 for CNN-LSTM; and 20 and 17 for mLSTM). BiLSTM-MA has the fewest SLA violations (5) in Workload 4, which is also less than all other models. This trend highlights the advantage of BiLSTM-MA in reducing SLA violations, which is significant for stable and predictable cloud services.

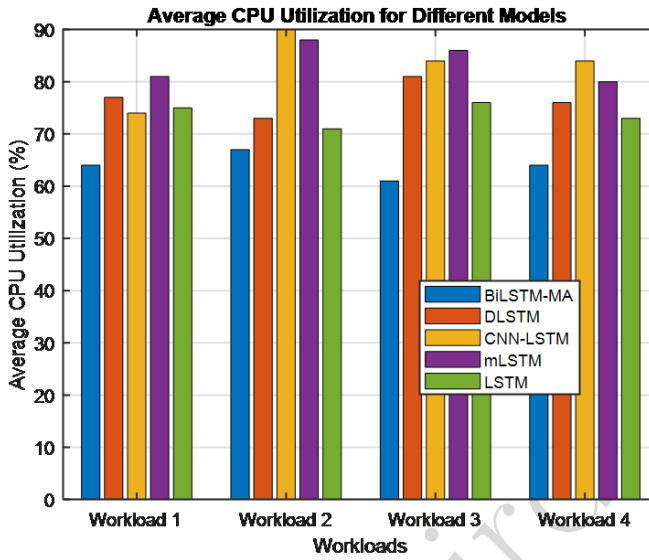


Figure 3. Average CPU Utilization analysis of proposed method and other methods

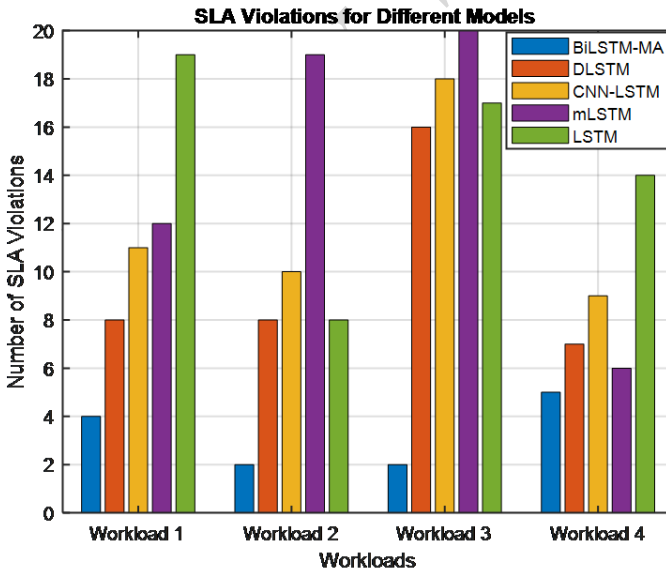


Figure 4. Comparing the SLA violations

Prediction accuracy measures: RMSE and RMSPE are error types often assessed in forecasting research. This was performed to determine the forecasting accuracy

and performance assessment of the predictions. It is evident that the calculations for both measures include comparing the goal values for the time series and projections that correspond to the target values for the time series. The findings produced by utilizing these two metrics provide numbers that are distinct from one another when computed. However, the relevance of using either metric in measuring the performance of prediction models is comparable. Because forecasting data tend to have varying scales in most instances, it is best to rely on RMSPE when estimating the relative error between various methods. This is because the RMSPE is a more precise method. The performance of various forecasting models can be evaluated using several tools, but the one most suited is the percentage error measure. This has been repeatedly proven in the literature. In addition, it provides the percentage error that can be used to evaluate the relative error between the various approaches, which is especially useful in situations in which time-series data samples have varying sizes. Consequently, we concentrated on these findings based on each method's RMSE and RMSPE measures.

Table 3
RMSE and RMPSE values of LSTM in different layers

Layers	Lag	Hidden units	Epochs	RMSE	RMPSE
1	3	[3]	1850	0.327	4.223
2	3	[3,2]	1284	0.312	3.781
3	3	[4,3,2]	1125	0.272	3.104
4	3	[5,4,3,2]	575	0.298	3.335

Table 4
RMSE and RMPSE values of mLSTM in different layers

Layers	Lag	Hidden units	Epochs	RMSE	RMPSE
1	3	[3]	1725	0.243	3.112
2	3	[3,2]	1840	0.283	3.438
3	3	[4,3,2]	982	0.297	3.499
4	3	[5,4,3,2]	780	0.258	3.233

Table 5
RMSE and RMPSE values of CNN-LSTM in different layers

Layers	Lag	Hidden units	Epochs	RMSE	RMPSE
1	3	[3]	1910	0.267	3.528
2	3	[3,2]	1122	0.231	3.212
3	3	[4,3,2]	1423	0.248	3.425
4	3	[5,4,3,2]	685	0.240	3.386

In the performance measure of lag 3, we can observe from Tables 6 and 7 that the best results for DLSTM and BiLSTM-MA were obtained by utilizing four layers

Table 6
RMSE and RMPSE values of DLSTM in different layers

Layers	Lag	Hidden units	Epochs	RMSE	RMPSE
1	3	[3]	1850	0.183	3.311
2	3	[3,2]	1820	0.135	3.118
3	3	[4,3,2]	1352	0.176	3.286
4	3	[5,4,3,2]	1278	0.156	3.212

Table 7
RMSE and RMPSE values of BiLSTM-MA in different layers

Layers	Lag	Hidden units	Epochs	RMSE	RMPSE
1	3	[3]	1980	0.178	3.287
2	3	[3,2]	1805	0.119	2.864
3	3	[4,3,2]	1524	0.146	3.177
4	3	[5,4,3,2]	1410	0.159	3.189

for various epochs, and the DLSTM first layer gave the best result. BiLSTM-MA outperformed all the other layers in the second layer. Tables 3, 4, and 5 show that LSTM, CNN, and CNN- LSTM achieved lower outcomes than the proposed method with the same lag value and layers. Based on the compiled comparison in Table 8, it is evident that the values of the proposed BiLSTM-MA method are the global minimum values compared with the other existing methods.

Table 8
Performance Summary of all methods

Methods	RMSE	RMSPE
LSTM	0.272	3.104
mLSTM	0.243	3.112
CNN-LSTM	0.231	3.212
DLSTM	0.135	3.118
BiLSTM-MA	0.119	2.864

Fig. 5 shows an overall accumulated comparison of BiLSTM-MA and the existing methods. The RMSE and RMSPE are two metrics used to accurately determine the value of each approach.

4.2. Discussion

To evaluate the efficacy of the various methods examined in this investigation, we conducted an analysis of the energy consumption, SLA violation, and CPU utilization across all four workloads of each deep learning method, including BiLSTM-MA, DLSTM, CNN-LSTM, mLSTM, and LSTM. Our findings indicate that the BiLSTM-MA method outperforms other methods in terms of energy consumption and SLA

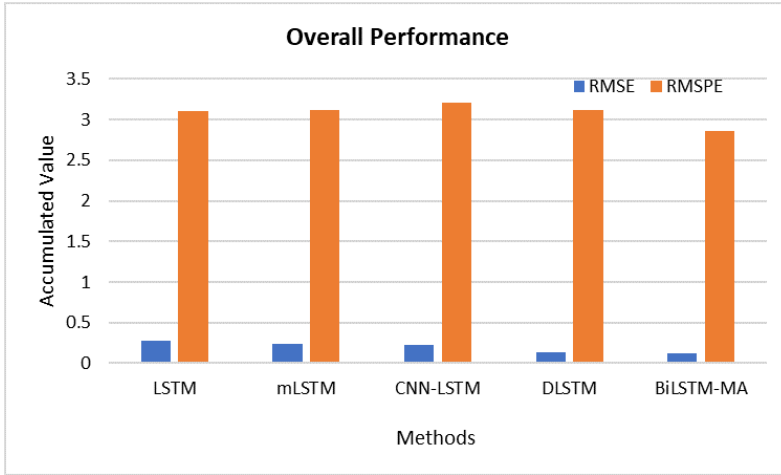


Figure 5. Overall accumulated comparison of all methods

violation metrics across all workloads. The BiLSTM-MA method exhibits superior performance in CPU utilization metrics for Workloads 1, 2, and 3 but underperforms in Workload 4 compared to alternative methods. Hence, it was observed that the BiLSTM-MA model exhibits limitations in its ability to perform effectively when confronted with a higher number of configurations necessary for complex systems. Table 8 presents the results of the BiLSTM-MA method and other deep learning techniques in terms of their performance on Google trace datasets. The results of our study indicate that our method has yielded a reduction in error rates across various metrics when compared to existing deep learning models such as DLSTM, CNN-LSTM, mLSTM, and LSTM. Specifically, our method demonstrated a decrease in RMSE error rates of 8.14%, 46%, 48%, and 55.39%, and a reduction in RMSPE error rates of 2.05%, 4.9%, 1.86%, and 2.17% on the datasets under consideration. Our model exhibits a temporal superiority of over two times compared with the other models, except for the LSTM model. The BiLSTM-MA model showed superior performance in terms of predictive accuracy compared to conventional deep learning techniques when applied to the dataset under consideration.

5. Conclusion

This paper proposes BiLSTM-MA, a new BiLSTM-based multi-modal self-attention model for workload prediction in cloud environments. Accurate workload prediction is crucial for minimizing service level agreement (SLA) violations and maximizing resource utilization. The proposed model outperformed existing predictive approaches in classifying complex cloud workload patterns extracted from CPU, memory, disk

I/O, and network traffic data.

Experimental results on the real-world Google Cloud Trace 2019 dataset show that BiLSTM-MA significantly improves several evaluation metrics compared with state-of-the-art models such as LSTM, CNN-LSTM, DLSTM, and mLSTM. BiLSTM-MA achieved the lowest RMSE and RMSPE values, indicating a higher prediction accuracy. The method demonstrates reductions in RMSE by 8.14%, 46%, 48%, and 55.39%, and reductions in RMSPE by 2.05%, 4.9%, 1.86%, and 2.17% over the existing methods.

Across all workloads, the experimental results show that the proposed BiLSTM-MA model consistently outperforms the other state-of-the-art models. It provides accurate workload predictions and effectively manages resources, as indicated by its lower average CPU usage, fewer SLA violations, and lower energy consumption. BiLSTM-MA enhances workload management, leading to improved resource utilization and reduced operational expenditure while maintaining service quality.

In future work, we plan to extend the proposed model to VM replacement and load prediction in cloud computing environments while incorporating additional parameters.

Funding

The authors declare that they have no known competing financial interests in this paper.

Competing interests

The authors declare that they have no known competing financial interests or personal relationships that could have influenced the work reported in this study.

Availability of data and material

The dataset used in this study is publicly available at: https://console.cloud.google.com/storage/browser/clusterdata_2019_a?authuser=0

References

- [1] Al-Sayed M.M.: Workload time series cumulative prediction mechanism for cloud resources using neural machine translation technique, *Journal of Grid Computing*, vol. 20(2), p. 16, 2022.
- [2] Chen W., Lu C., Ye K., Wang Y., Xu C.Z.: Rptcn: Resource prediction for high-dynamic workloads in clouds based on deep learning. In: *2021 IEEE International Conference on Cluster Computing (CLUSTER)*, pp. 59–69, IEEE, 2021.
- [3] Devi K.L., Valli S.: Time series-based workload prediction using the statistical hybrid model for the cloud environment, *Computing*, vol. 105(2), pp. 353–374, 2023.

- [4] Ding W., Luo F., Han L., Gu C., Lu H., Fuentes J.: Adaptive virtual machine consolidation framework based on performance-to-power ratio in cloud data centers, *Future Generation Computer Systems*, vol. 111, pp. 254–270, 2020.
- [5] Garg R., Arora I., Gupta A.: Predictive VM placement algorithm for resource optimization: leveraging deep learning forecasting and resource relationship modeling, *International Journal of Computers and Applications*, vol. 47(4), pp. 398–411, 2025.
- [6] Ghasemi A., Toroghi Haghighat A.: A multi-objective load balancing algorithm for virtual machine placement in cloud data centers based on machine learning, *Computing*, vol. 102(9), pp. 2049–2072, 2020.
- [7] Jeong B., Baek S., Park S., Jeon J., Jeong Y.S.: Stable and efficient resource management using deep neural network on cloud computing, *Neurocomputing*, vol. 521, pp. 99–112, 2023.
- [8] Jing Y., Shen L., Yao C., Fan F., Wang X.: Hlem-ymp: An effective virtual machine placement algorithm for minimizing sla violations in cloud data centers, *IEEE Systems Journal*, 2024.
- [9] Kshatriya D., Lepakshi V.A.: SLA Aware Optimized Task Scheduling Model for Faster Execution of Workloads Among Federated Clouds, *Wireless Personal Communications*, vol. 135(3), pp. 1635–1661, 2024.
- [10] Kumar J., Goomer R., Singh A.K.: Long short term memory recurrent neural network (LSTM-RNN) based workload forecasting model for cloud datacenters, *Procedia computer science*, vol. 125, pp. 676–682, 2018.
- [11] Kumar J., Saxena D., Kumar J., Bhadoria A., Anjali A., Singh A.K.: A hybrid neural network and cooperative PSO model for dynamic cloud workloads prediction, *Computing*, vol. 107(3), pp. 1–48, 2025.
- [12] Lilhore U.K., Simaiya S., Guleria K., Prasad D.: An efficient load balancing method by using machine learning-based VM distribution and dynamic resource mapping, *Journal of Computational and Theoretical Nanoscience*, vol. 17(6), pp. 2545–2551, 2020.
- [13] Liu J., Tan X., Wang Y.: CSSAP: Software aging prediction for cloud services based on ARIMA-LSTM hybrid model. In: *2019 IEEE International Conference on Web Services (ICWS)*, pp. 283–290, IEEE, 2019.
- [14] Naveen Kumar M., Annappa B., Yadav V.: Efficient Kalman filter based deep learning approaches for workload prediction in cloud and edge environments, *Computing*, vol. 107(1), p. 10, 2025.
- [15] Nehra P., Kesswani N.: A workload prediction model for reducing service level agreement violations in cloud data centers, *Decision Analytics Journal*, vol. 11, p. 100463, 2024.
- [16] Ouham S., Hadi Y., Ullah A.: An efficient forecasting approach for resource utilization in cloud data center using CNN-LSTM model, *Neural Computing and Applications*, vol. 33(16), pp. 10043–10055, 2021.

- [17] Pattanayak R.M., Panigrahi S., Behera H.: High-order fuzzy time series forecasting by using membership values along with data and support vector machine, *Arabian Journal for Science and Engineering*, vol. 45(12), pp. 10311–10325, 2020.
- [18] Pawar N., Lilhore U.K., Agrawal N.: A hybrid ACHBDF load balancing method for optimum resource utilization in cloud computing, *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, vol. 3307, pp. 367–373, 2017.
- [19] Sabyasachi A.S., Sahoo B.M., Ranganath A.: Deep cnn and lstm approaches for efficient workload prediction in cloud environment, *Procedia Computer Science*, vol. 235, pp. 2651–2661, 2024.
- [20] Sagheer A., Kotb M.: Time series forecasting of petroleum production using deep LSTM recurrent networks, *Neurocomputing*, vol. 323, pp. 203–213, 2019.
- [21] Sharma M., Garg R.: An artificial neural network based approach for energy efficient task scheduling in cloud data centers, *Sustainable Computing: Informatics and Systems*, vol. 26, p. 100373, 2020.
- [22] Simaiya S., Gautam V., Lilhore U.K., Garg A., Ghosh P., Trivedi N.K., Anand A.: EEPSA: Energy efficiency priority scheduling algorithm for cloud computing. In: *2021 2nd international conference on smart electronics and communication (ICOSEC)*, pp. 1064–1069, IEEE, 2021.
- [23] Xu M., Song C., Wu H., Gill S.S., Ye K., Xu C.: esDNN: deep neural network based multivariate workload prediction in cloud computing environments, *ACM Transactions on Internet Technology (TOIT)*, vol. 22(3), pp. 1–24, 2022.
- [24] Yazdani P., Sharifian S.: E2LG: a multiscale ensemble of LSTM/GAN deep learning architecture for multistep-ahead cloud workload prediction, *The Journal of Supercomputing*, vol. 77(10), pp. 11052–11082, 2021.
- [25] Zhu Y., Zhang W., Chen Y., Gao H.: A novel approach to workload prediction using attention-based LSTM encoder-decoder network in cloud environment, *EURASIP Journal on Wireless Communications and Networking*, vol. 2019(1), p. 274, 2019.

Affiliations

Abhishek Kumar Pandey

Department of Information Technology and Computer Application, Madan Mohan Malaviya University of Technology, Gorakhpur, Uttar Pradesh, India, 273010, abhishekpmmmut6@gmail.com

Sarvpal Singh

Department of Information Technology and Computer Application, Madan Mohan Malaviya University of Technology, Gorakhpur, Uttar Pradesh, India, 273010, spsitca@mmmut.ac.in

Received: 06.05.2025

Revised: 26.12.2025

Accepted: 14.09.2026