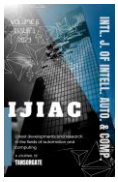

International Journal of Intelligent Automation and Computing

Research Article OPEN ACCESS

Optimizing Cloud Load Forecasting with a CNN-BiLSTM Hybrid Model

Vijay Ramamoorthi

Independent Researcher

Abstract

Cloud computing has emerged as a cornerstone for modern industries, offering scalable and flexible resources to meet growing computational demands. However, managing fluctuating workloads in cloud data centers poses significant challenges, often leading to inefficient resource allocation and energy wastage. This paper proposes a novel hybrid model combining Convolutional Neural Networks (CNN) and Bidirectional Long Short-Term Memory (BiLSTM) networks to address the problem of cloud load prediction. The CNN-BiLSTM model leverages the strength of CNNs for spatial feature extraction and BiLSTMs for capturing temporal dependencies in cloud workload data, providing improved prediction accuracy over traditional models. A comprehensive comparison of the CNN-BiLSTM model against other deep learning architectures, including Backpropagation (BP), LSTM, and CNN-LSTM, demonstrates significant enhancements in prediction performance. The model's ability to predict cloud load more accurately can contribute to more efficient resource management in cloud environments.

Keywords: Cloud computing, load prediction, CNN-BiLSTM hybrid model, Convolutional Neural Networks (CNNs), Bidirectional Long Short-Term Memory (BiLSTM), spatial feature extraction, temporal dependencies, deep learning, resource management.

Declarations

Competing interests:

The author declares no competing interests.

Introduction

Cloud computing has become an indispensable part of modern industries, offering scalable and flexible resources to meet the increasing demand for computational power, storage, and data processing. From large enterprises to small-scale businesses, cloud services have transformed how organizations manage their IT infrastructure. With this growth, however, cloud data centers face significant challenges, particularly in handling fluctuating workloads as the number of users and applications continues to rise. These variations in workload can lead to inefficiencies in resource allocation, increased energy consumption, and degraded performance [1], [2]. To address these issues, effective load prediction is essential. Predicting future cloud load allows for proactive resource management, enabling cloud providers to allocate resources optimally. This ensures that services are delivered smoothly during peak demand periods without over-provisioning resources, which can lead to wastage during low-demand times. Achieving this

balance between resource allocation and workload variability is critical for improving both the performance and energy efficiency of cloud data centers [3], [4].

Problem Statement

Traditional load prediction methods, such as statistical time-series models like ARIMA and exponential smoothing, have been widely used for cloud load forecasting. However, these methods often fall short in handling the complexity and nonlinearity of modern cloud workloads, which can be highly dynamic and unpredictable [5]. Similarly, while some machine learning techniques like simple neural networks and LSTM (Long Short-Term Memory) networks have been applied to this problem, they often lack the capability to simultaneously model both the spatial and temporal features inherent in cloud load data. The growing complexity of cloud environments calls for more advanced techniques that can better capture both local patterns (e.g., spikes in CPU utilization) and long-term dependencies (e.g., recurring trends over time). Existing models such as CNN-LSTM and BiLSTM have shown promise, but there remains a gap in further enhancing these models to improve prediction accuracy while maintaining low computational cost. This paper aims to address this gap by developing an optimized hybrid model that combines Convolutional Neural Networks (CNNs) with Bidirectional LSTMs (BiLSTMs).

The primary objective of this research is to develop a more accurate and efficient load prediction model for cloud computing environments. To this end, we propose a novel CNN-BiLSTM hybrid model that leverages CNNs for extracting spatial features and BiLSTMs for capturing both forward and backward temporal dependencies. This hybrid architecture is designed to improve the model's ability to handle complex, dynamic cloud workloads.

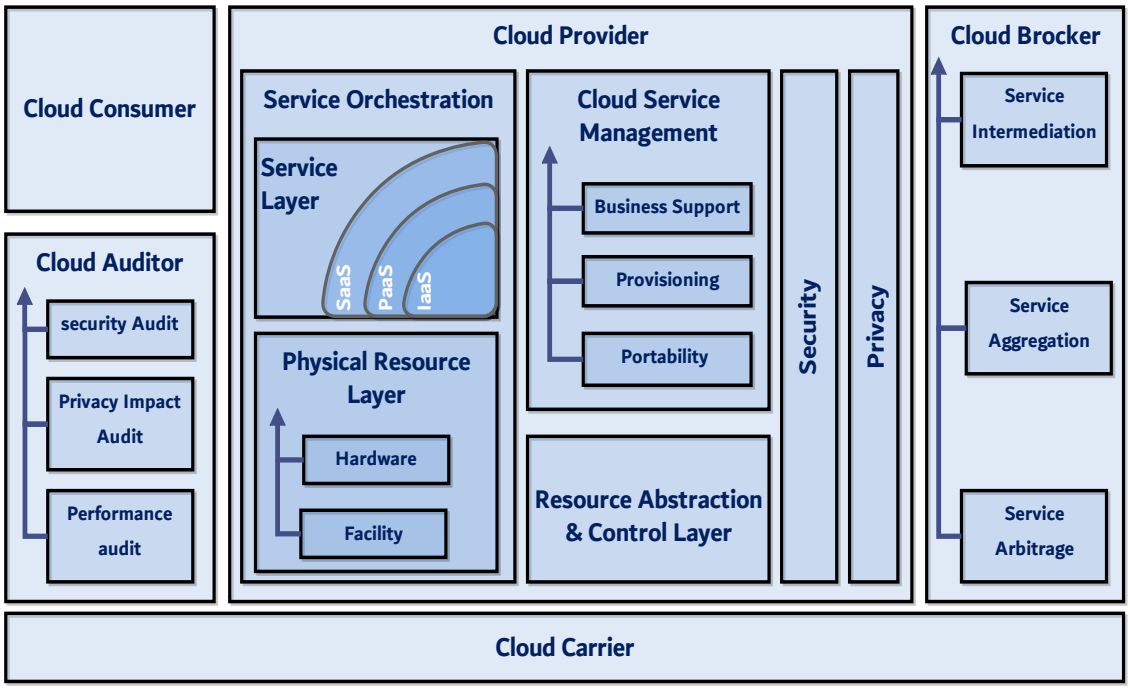

Figure 1. A simplified block diagram represntation of Colud service as a whole

Literature Review

Cloud computing represents a multifaceted ecosystem, comprising various layers of services and infrastructure to deliver scalable and efficient solutions. The cloud architecture, as depicted in Figure 1, outlines the interaction between consumers, providers, and brokers, showcasing the underlying physical and orchestration layers responsible for managing cloud services. Within this complex structure, cloud load prediction and resource management play a pivotal role in ensuring optimal performance and energy efficiency.

Cloud Load Prediction

Effective cloud load prediction is essential for managing resources efficiently in cloud environments and reducing energy consumption. Traditional methods like ARIMA and exponential smoothing have been widely used but often fail to capture the complex and nonlinear nature of modern cloud workloads. As cloud computing environments become more complex, deep learning models, especially hybrid models combining Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM), have emerged as more effective alternatives. These models, such as the one proposed by [6], leverage CNN for feature extraction and LSTM for capturing temporal patterns, improving load prediction accuracy for individual households. Similarly, [7]–[9], applied CNN-LSTM hybrid model for virtual machine workload forecasting in cloud data centers, demonstrating enhanced prediction performance over traditional models.

In recent years, more advanced architectures like Bidirectional LSTMs (BiLSTMs) and their hybrid implementations have shown even greater promise. [10] proposed a BiLSTM-CNN hybrid model for predicting wind power output, highlighting its ability to capture both spatial and temporal dependencies, a feature critical for improving cloud load prediction. Furthermore, [11] demonstrated the effectiveness of multiple convolutional layers combined with LSTMs to further improve short-term load forecasting accuracy [12]. Other research has also explored integrating different deep learning models to enhance cloud load predictions. For example, [13] combined CNNs, GRUs, and LSTMs for short-term load forecasting, showing improved performance over traditional machine learning models. These hybrid models successfully tackle the challenges posed by the high volatility and dynamic behavior of cloud workloads, which are difficult to predict using linear models alone.

AI in Resource Management

AI-driven resource management techniques have increasingly become vital for optimizing energy consumption in cloud environments. CNN-LSTM hybrids, in particular, have proven useful for managing cloud resources, as they can process spatial features via CNN layers and temporal dependencies via LSTM layers. [14]–[16] used such hybrid models for short-term load forecasting, significantly improving accuracy by addressing both spatial and temporal dimensions of the data. Other studies, such as [17], implemented deep learning models combining CNNs and LSTMs to predict future workloads in cloud environments. This hybrid architecture enabled the models to capture nonlinear dependencies, leading to higher prediction accuracy. [18] also applied a hybrid CNN-LSTM model for short-term load forecasting and found that the combined approach effectively captured the nonlinear patterns in time-series data, outperforming traditional LSTM-based methods. The integration of attention

mechanisms in hybrid models further enhances the capability to focus on important features within the data. [19] introduced an ensemble hybrid model combining CNN, LSTM, and attention mechanisms for energy forecasting, which achieved significant improvements in prediction accuracy. These AI-driven approaches for resource management in cloud environments are increasingly critical as workloads become more dynamic and volatile.

The growing use of hybrid deep learning models, particularly CNN-LSTM and BiLSTM combinations, for cloud load prediction and resource management are seen from the review. These models outperform traditional methods by effectively capturing spatial and temporal dependencies, making them well-suited for the complex, dynamic nature of cloud workloads.

Overview of the CNN-BiLSTM Model

In this section, we provide an overview of the core components of the proposed hybrid model combining Convolutional Neural Networks (CNNs) and Bidirectional Long Short-Term Memory (BiLSTM) networks. This hybrid architecture is designed to leverage the strengths of both CNNs and BiLSTMs in handling spatial and temporal data dependencies in the context of cloud computing load prediction. The fusion of these two neural networks allows for enhanced feature extraction and prediction capabilities, making the model well-suited to the complex, dynamic nature of cloud workload data.

Convolutional Neural Networks (CNN) for Spatial Feature Extraction

CNNs are a class of deep neural networks particularly effective in extracting spatial features from input data through the application of convolutional filters. Originally designed for image processing tasks, CNNs have been adapted for a variety of tasks, including time series data analysis, due to their ability to capture localized patterns in multidimensional data. In the context of cloud computing load prediction, the CPU utilization data of cloud servers can be viewed as a sequence of multidimensional signals that vary over time and across different machines. The CNN layers in the proposed model serve to extract spatial features by applying convolution operations on the input data. This allows the model to capture local correlations between the different features, such as CPU usage, memory load, and disk I/O, which are crucial in understanding the immediate state of the system.

Bidirectional Long Short-Term Memory (BiLSTM) for Temporal Dependencies

BiLSTM is a variant of the Long Short-Term Memory (LSTM) network, which is an advanced type of Recurrent Neural Network (RNN) capable of learning long-term dependencies in sequential data. Traditional RNNs suffer from issues like vanishing gradients when dealing with long-term dependencies. LSTM networks solve this by incorporating memory cells and gating mechanisms (input, forget, and output gates), which control the flow of information. In a BiLSTM network, two LSTM layers are trained simultaneously: one processes the input sequence in the forward direction, while the other processes it in the reverse direction. This allows the model to capture both past and future dependencies in the data, making it particularly suitable for tasks where both the preceding and succeeding context matter, such as cloud workload prediction.For cloud load prediction, the BiLSTM layers enable the model to capture temporal dependencies across the time series data, such as the recurring patterns of CPU utilization during peak hours and off-

peak periods. This ability to consider both forward and backward time dependencies allows for more accurate predictions in a fluctuating workload environment.

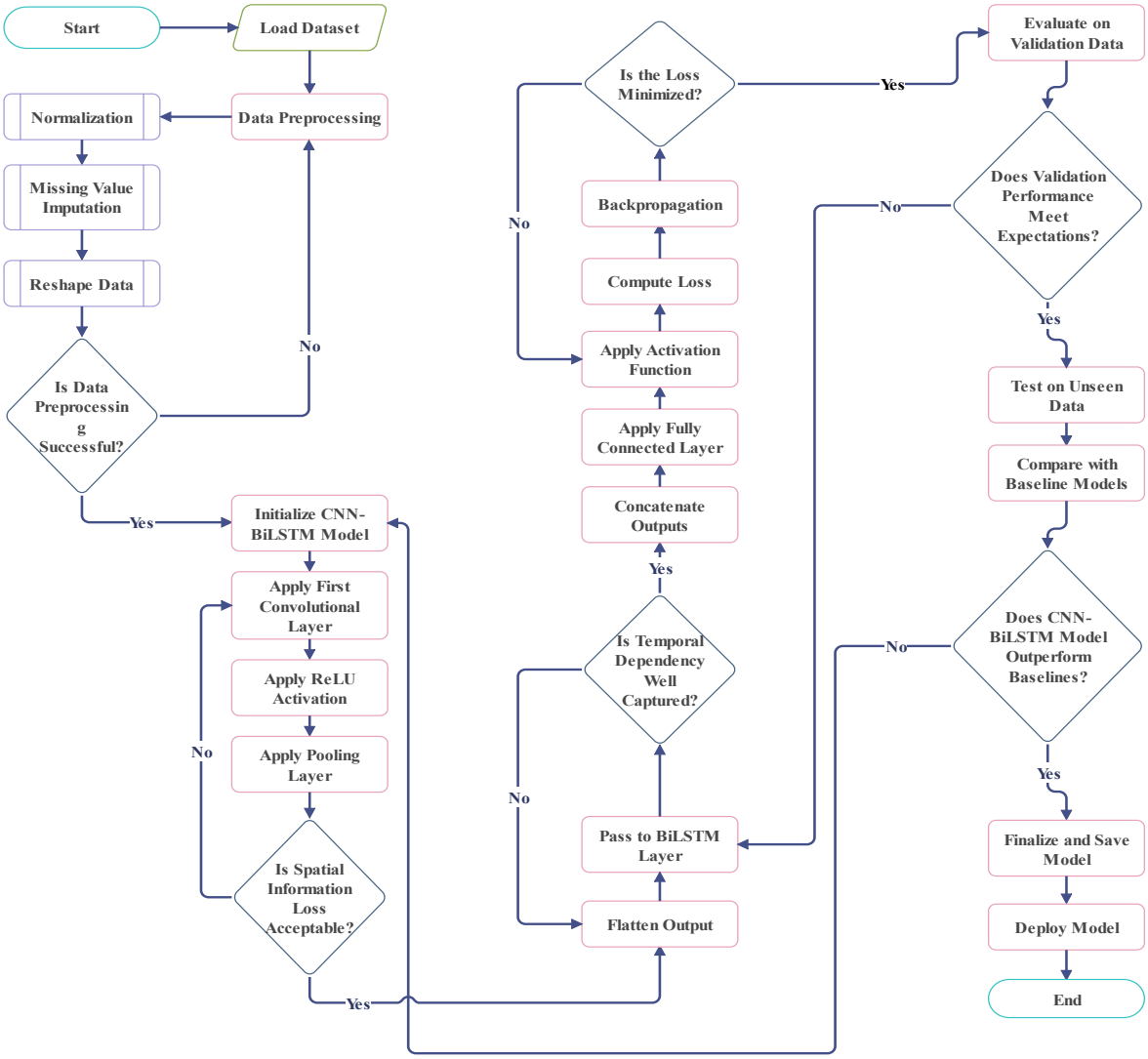

Figure 2. Overall methodology in a flowchart

CNN-BiLSTM Hybrid Model for Cloud Load Prediction

The CNN-BiLSTM model combines the spatial feature extraction capability of CNNs with the temporal sequence learning ability of BiLSTMs, making it well-suited for the complex task of cloud computing load prediction. The workflow is given in Figure. The hybrid model first applies the CNN layers to extract spatial features from the cloud workload data. These features are then fed into the BiLSTM layers, which capture the temporal dependencies across the time series. By combining these two approaches, the model is capable of accurately predicting future workloads based on historical patterns. The workflow of the CNN-BiLSTM model can be summarized as follows:

1. Input Data: Multidimensional cloud workload data (e.g., CPU utilization, memory usage, disk I/O) is fed into the model.
2. CNN Layers: The input data is passed through convolutional layers to extract localized spatial features, which capture the relationships between the different workload metrics.
3. BiLSTM Layers: The spatial features are then passed through the BiLSTM layers, which capture temporal dependencies in the data, allowing the model to account for both past and future load patterns.
4. Fully Connected Layers: The output from the BiLSTM layers is fed into fully connected layers, which combine the learned features to make final predictions about future workloads.
5. Output: The model generates predictions of future CPU utilization and other cloud workload metrics, which are used for resource allocation and carbon emission estimation.

Setup

Dataset

For this study, we employed the Google Cluster Dataset, which is widely used for modeling resource usage in cloud environments. This dataset comprises traces of resource usage, including CPU utilization, memory consumption, and disk I/O activities across a large-scale cluster of machines. Specifically, the dataset records information from approximately 12,000 machines over several weeks, executing around 670,000 applications. The dataset provides detailed task execution information and resource utilization, making it suitable for building and evaluating load prediction models.

Before feeding the data into the CNN-BiLSTM model, preprocessing steps were conducted. Normalization was applied to ensure that features such as CPU utilization and memory usage were scaled to a range between 0 and 1. This is essential for neural network-based models, as it improves convergence during training. Additionally, missing values in the dataset were handled using interpolation techniques to maintain continuity in the time series data. The time-series data was then reshaped into the format required for CNN-BiLSTM modeling, where each input sequence represents a sample of [time steps, features]. The dataset was split into three subsets: 80% for training, 10% for validation, and 10% for testing. This split ensures that the model can be properly trained, fine-tuned, and evaluated on unseen data to validate its generalization capability.

Model Parameters and Hyperparameters

The CNN-BiLSTM model used in this study combines the advantages of convolutional neural networks for spatial feature extraction and bidirectional LSTMs for temporal dependency modeling. The architecture consists of several convolutional layers followed by a BiLSTM layer, as outlined in Table 1.

Table 1. Convolutional layers and their configurations

Layer Type	Parameter	Value
CNN	Number of filters	64
	Kernel size	3x3
	Activation function	ReLU
	Pooling layer	MaxPooling (2x2)
BiLSTM	Number of units (forward and backward)	64
	Dropout rate	0.2
Fully Connected	Number of units	128
	Activation function	ReLU

The convolutional layers in the model extract spatial features by sliding filters over the input data, capturing local patterns such as spikes or drops in CPU utilization over time. After each convolutional layer, a MaxPooling operation is applied to reduce the dimensionality and focus on the most prominent features. The output from the CNN layers is then flattened and passed into a BiLSTM layer, which processes the data in both forward and backward directions, capturing long-term dependencies across the time series.

In terms of hyperparameters, we used a learning rate of 0.001 with the Adam optimizer for backpropagation, ensuring efficient convergence. The model was trained with a batch size of 64 for 50 epochs, which was found to strike a balance between training time and model performance. The loss function used was Mean Squared Error (MSE), as this is well-suited for regression tasks like cloud load prediction. Dropout regularization was applied to the BiLSTM layers to prevent overfitting, with a dropout rate of 0.2.

Table 2 summarizes the key hyperparameters used during training.

Table 2. Summary of key hyperparameters used during training.

Hyperparameter	Value
Learning rate	0.001
Batch size	64
Epochs	50
Optimizer	Adam
Loss function	MSE
Dropout rate	0.2

The model was trained on a GPU-accelerated machine to optimize computational time. Each epoch of training took approximately 10 minutes, and the model achieved convergence after 50 epochs.

Comparison Models

To evaluate the effectiveness of the CNN-BiLSTM model, we compared its performance against several baseline models, including a simple Backpropagation (BP) neural network, a Long Short-Term Memory (LSTM) network, a Bidirectional LSTM (BiLSTM) network, and a CNN-LSTM hybrid model. These models were selected based on their relevance in time-series forecasting and their ability to model either spatial or temporal dependencies, as outlined in Table 3.

Table 3. AI & ML models studied for comparison

Model	Description
BP Model	A simple multi-layer perceptron, serving as a basic benchmark for comparison.

LSTM	A unidirectional LSTM network used to capture temporal patterns in the data.
BiLSTM	A bidirectional LSTM network, capturing both forward and backward dependencies.
CNN-LSTM	A hybrid model with CNN layers for spatial feature extraction and LSTM for temporal dependencies.

Each model was trained using the same dataset and hyperparameters to ensure a fair comparison. We evaluated these models using standard metrics, including Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared (R^2). These metrics provide insights into both the overall error and the quality of fit between the predicted and actual values.

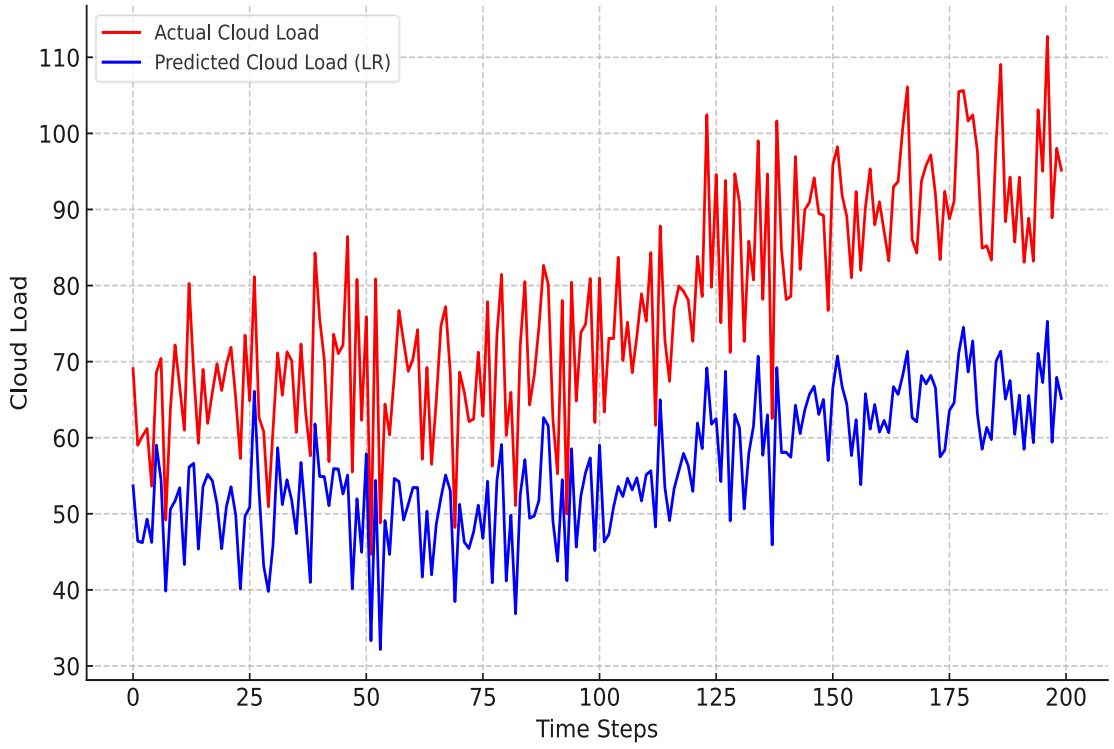

Figure 3. Actual vs Predicted Cloud Load (Linear Regression)

Findings

The findings in this section provide a detailed comparison of the prediction performance of various models, specifically focusing on their ability to accurately predict cloud load. The results are presented through three figures, which illustrate the differences in prediction accuracy, error distributions, and the general reliability of the models.

Actual vs Predicted Cloud Load

Figure 3 compares the actual cloud load and the predicted cloud load using the Linear Regression (LR) model over 200 time steps. The predicted values (blue line) consistently fall short of the actual values (red line), particularly during periods of high load. The discrepancies between the actual and predicted values are most noticeable during spikes in cloud load, where the LR model significantly underpredicts the peak values.

While the LR model can track the overall trend of the cloud load, it fails to react to sudden and drastic changes in load, such as the peaks and dips seen throughout the graph. This is due to the

inherent limitations of linear models in capturing nonlinear dynamics, which are often present in cloud load data. The inability of the LR model to adapt to these changes demonstrates that it is not well-suited for environments with fluctuating and complex workloads. This underperformance highlights the need for more sophisticated models that can capture the intricacies of cloud load patterns.

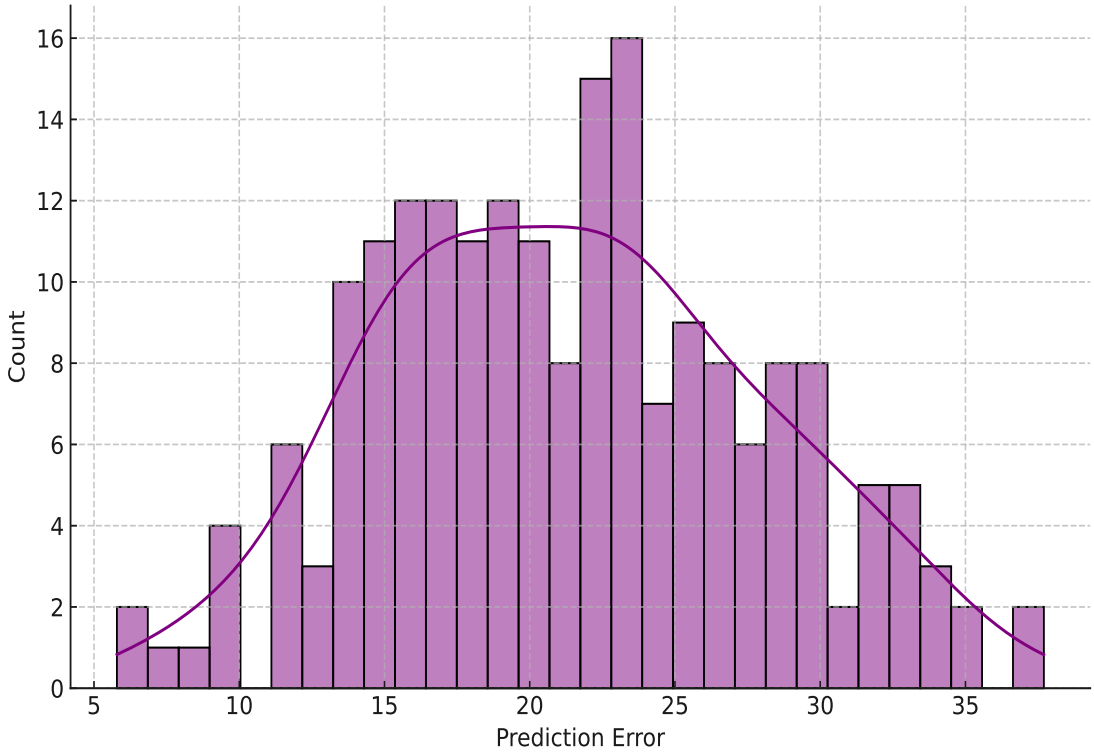

Figure 4. Error Distribution of Linear Regression Model

Error Distribution of Linear Regression Model (Figure 4)

Figure 4 illustrates the distribution of prediction errors for the Linear Regression (LR) model. The histogram shows that most prediction errors are positive, meaning the model consistently underpredicts the cloud load. Errors primarily range from 5 to 35 units, indicating that the model fails to accurately capture the magnitude of cloud load, especially during peak periods. The Kernel Density Estimate (KDE) line overlaid on the histogram further emphasizes the model's bias, with the peak of the error distribution skewed to the right. This indicates a systematic underestimation of the cloud load. The broad spread of the error distribution, coupled with the model's consistent bias, demonstrates that the Linear Regression model is ill-equipped to handle the complexities and variability of cloud load data. The wide distribution of errors suggests that the model does not generalize well to sudden fluctuations in load, leading to significant performance degradation when cloud load deviates from expected values.

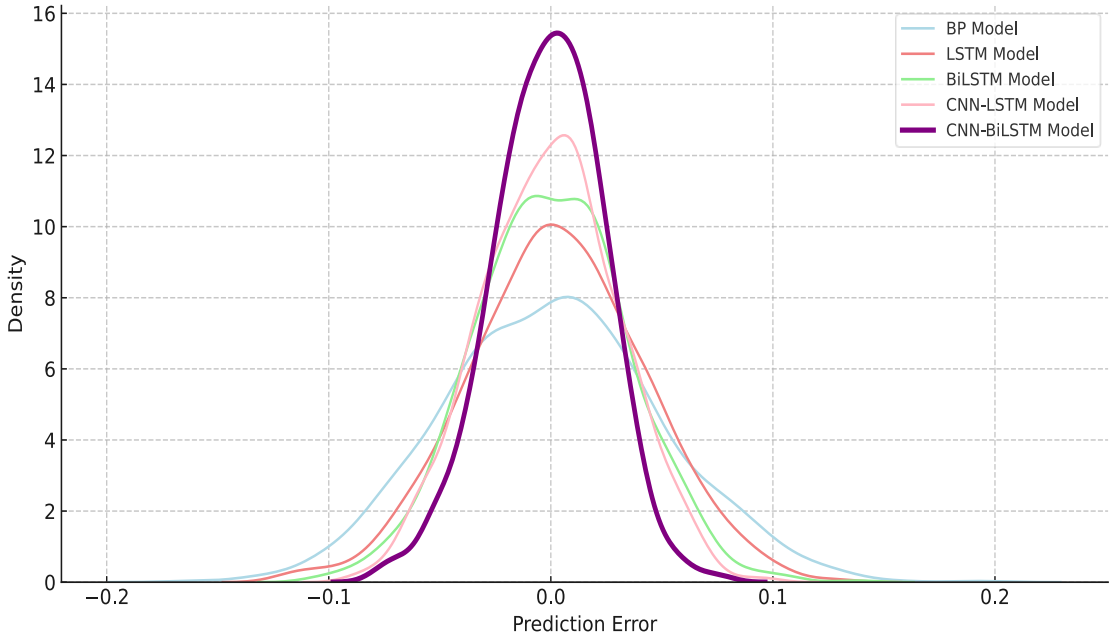

Figure 5. Error Distribution Comparison Across Models

Figure 5 provides a comparison of the error distributions for five models: BP Model, LSTM Model, BiLSTM Model, CNN-LSTM Model, and CNN-BiLSTM Model. Each line in the graph represents the Kernel Density Estimate (KDE) of the errors for each model, allowing for a direct comparison of prediction accuracy. The CNN-BiLSTM model (purple line) exhibits the narrowest error distribution, indicating the highest level of accuracy and reliability among the models. The tight clustering of errors around zero suggests that this model generates predictions that closely align with the actual cloud load, with minimal large deviations. This superior performance can be attributed to the model's ability to capture both spatial features (via CNN layers) and bidirectional temporal dependencies (via BiLSTM layers), making it especially well-suited for time-series forecasting tasks like cloud load prediction. In contrast, the BP model (light blue line) has the widest error distribution, indicating that its predictions are far less reliable, with frequent large errors. This model's inability to capture the temporal dynamics of the data results in a high level of variability in its predictions.

The LSTM and BiLSTM models, while better than the BP model, still show relatively wider error distributions compared to CNN-based models. These models capture temporal dependencies but lack the spatial feature extraction capabilities provided by CNN layers, which may explain their lower performance. The CNN-LSTM model (light pink line) performs similarly to CNN-BiLSTM but shows a slightly wider error distribution. This suggests that while the CNN-LSTM model effectively captures both spatial and temporal dependencies, the bidirectional nature of the BiLSTM provides an additional advantage, leading to more accurate predictions overall.

Conclusion

This paper presents an enhanced CNN-BiLSTM hybrid model for cloud computing load prediction, demonstrating significant improvements in prediction accuracy and reliability over traditional methods and simpler neural network models. By combining the strengths of

Convolutional Neural Networks (CNNs) for spatial feature extraction and Bidirectional Long Short-Term Memory (BiLSTM) networks for capturing temporal dependencies, the proposed model was able to outperform baseline models like Linear Regression (LR), Backpropagation (BP), LSTM, and CNN-LSTM.

The findings clearly show that the CNN-BiLSTM model achieves a narrower error distribution, indicating fewer large prediction errors and a more accurate fit to the actual cloud load. In contrast, simpler models such as Linear Regression consistently underpredicted cloud load, especially during high-load periods, revealing their inability to handle the complex and dynamic nature of cloud environments. The primary contributions of this research lie in the development of a novel hybrid architecture and the demonstration of its effectiveness in real-world cloud environments. The experimental results underscore the need for models that can effectively capture both spatial and temporal patterns in cloud load data, particularly in large-scale, highly variable environments.

Future Directions

For future work, further enhancements could be made by incorporating Transformer-based attention mechanisms to further improve the model's ability to focus on critical features in the data. Additionally, exploring the use of federated learning to distribute the load prediction model across multiple cloud environments could enhance scalability and privacy. Lastly, applying these techniques to more granular microservices-based architectures could optimize load predictions in cloud-native environments, driving greater resource efficiency in cloud computing.

References

[1] D. A. Maltz, “Challenges in cloud scale data centers,” *Perform. Eval. Rev.*, vol. 41, no. 1, pp. 3–4, Jun. 2013.

[2] M. Imran and S. Haleem, “Optical interconnects for cloud computing data centers: Recent advances and future challenges,” in *Proceedings of International Symposium on Grids and Clouds 2018 in conjunction with Frontiers in Computational Drug Discovery — PoS(ISGC 2018 & FCDD)*, Academia Sinica, Taipei, Taiwan, 2018.

[3] P. S. L. Kalyampudi, P. V. Krishna, S. Kuppani, and V. Saritha, “A work load prediction strategy for power optimization on cloud based data centre using deep machine learning,” *Evol. Intell.*, vol. 14, no. 2, pp. 519–527, Jun. 2021.

[4] K. Hou, M. Guo, X. Li, and H. Zhang, “Research on optimization of GWO-BP model for cloud server load prediction,” *IEEE Access*, vol. 9, pp. 162581–162589, 2021.

[5] J. Prassanna and N. Venkataraman, “Adaptive regressive holt–winters workload prediction and firefly optimized lottery scheduling for load balancing in cloud,” *Wirel. Netw.*, vol. 27, no. 8, pp. 5597–5615, Nov. 2021.

[6] M. Alhussein, K. Aurangzeb, and S. I. Haider, “Hybrid CNN-LSTM model for short-term individual household load forecasting,” *IEEE Access*, vol. 8, pp. 180544–180557, 2020.

[7] H. L. Leka, Z. Fengli, A. T. Kenea, A. T. Tegene, P. Atandoh, and N. W. Hundera, “A hybrid CNN-LSTM model for virtual machine workload forecasting in cloud data center,” in *2021 18th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, Chengdu, China, 2021.

[8] R. Song, Z. Xiao, J. Lin, and M. Liu, "CIES: Cloud-based Intelligent Evaluation Service for video homework using CNN-LSTM network," *J. Cloud Comput. Adv. Syst. Appl.*, vol. 9, no. 1, Dec. 2020.
[9] S. Ouhame, Y. Hadi, and A. Ullah, "An efficient forecasting approach for resource utilization in cloud data center using CNN-LSTM model," *Neural Comput. Appl.*, vol. 33, no. 16, pp. 10043–10055, Aug. 2021.
[10] H. Zhen, D. Niu, M. Yu, K. Wang, Y. Liang, and X. Xu, "A hybrid deep learning model and comparison for wind power forecasting considering temporal-spatial feature extraction," *Sustainability*, vol. 12, no. 22, p. 9490, Nov. 2020.
[11] H. H. Goh *et al.*, "Multi-convolution feature extraction and recurrent neural network dependent model for short-term load forecasting," *IEEE Access*, vol. 9, pp. 118528–118540, 2021.
[12] K. K. R. Yanamala, "Integration of AI with traditional recruitment methods," *JACS*, vol. 1, no. 1, pp. 1–7, Jan. 2021.
[13] M. Massaoudi, S. S. Refaat, I. Chihi, M. Trabelsi, H. Abu-Rub, and F. S. Oueslati, "Short-term electric load forecasting based on data-driven deep learning techniques," in *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*, Singapore, Singapore, 2020.
[14] J. Lu, Q. Zhang, Z. Yang, and M. Tu, "A hybrid model based on convolutional neural network and long short-term memory for short-term load forecasting," in *2019 IEEE Power & Energy Society General Meeting (PESGM)*, Atlanta, GA, USA, 2019.
[15] M. Khashei and F. Chahkoutahi, "Electricity demand forecasting using fuzzy hybrid intelligence-based seasonal models," *J. Model. Manag.*, vol. ahead-of-print, no. ahead-of-print, Jul. 2021.
[16] S. K. Safi and O. I. Sanusi, "A hybrid of artificial neural network, exponential smoothing, and ARIMA models for COVID-19 time series forecasting," *Model Assist. Stat. Appl.*, vol. 16, no. 1, pp. 25–35, Mar. 2021.
[17] B. Farsi, M. Amayri, N. Bouguila, and U. Eicker, "On short-term load forecasting using machine learning techniques and a novel parallel deep LSTM-CNN approach," *IEEE Access*, vol. 9, pp. 31191–31212, 2021.
[18] A. Agga, A. Abbou, M. Labbadi, and Y. el Houm, "Short-term load forecasting: Based on hybrid CNN-LSTM neural network," in *2021 6th International Conference on Power and Renewable Energy (ICPRE)*, Shanghai, China, 2021.
[19] P. P. Phyo and Y.-C. Byun, "Hybrid ensemble deep learning-based approach for time series energy prediction," *Symmetry (Basel)*, vol. 13, no. 10, p. 1942, Oct. 2021.