Journal of Computer Technology and Software

ISSN:2998-2383

Vol. 4, No. 7, 2025

Multi-Scale Transformer for CPU Utilization Time Series Forecasting in Cloud Computing

Zilong Wang[1*], Shengsheng Lin[2]
[1]New York University, New York, USA
[2]University of Texas at Dallas, Richardson, USA
*Corresponding author: Zilong Wang; zilongwang086@gmail.com

Abstract: This paper addresses the complexity and challenges of CPU utilization prediction in cloud computing environments and proposes a time series modeling method based on a multi-scale Transformer. The method first preprocesses the input utilization sequence and performs window segmentation. It then constructs embeddings at multiple scales to capture both short-term fluctuations and long-term trends. Next, the model applies a multi-head self-attention mechanism to effectively capture global dependencies in the sequence. Cross-scale interaction and weighted fusion strategies are further introduced to enhance the comprehensive representation of multi-level dynamic features. In the prediction stage, the model uses the fused feature representations to estimate future CPU utilization. Training is performed with mean squared error as the optimization objective to ensure stability and accuracy. The experimental study includes multiple comparative analyses and sensitivity tests, covering mainstream methods such as LSTM, BILSTM, 1DCNN, and Transformer. The results show that the proposed model achieves better performance in terms of mean squared error, root mean squared error, mean absolute error, and the coefficient of determination. Further sensitivity experiments on hyperparameters and the multi-scale mechanism verify the effectiveness and adaptability of the method in complex cloud computing environments. The overall results demonstrate the advantages and value of the multi-scale Transformer in CPU utilization prediction tasks.

Keywords: Cloud computing resource management, multi-scale modeling, time series prediction, Transformer

1. Introduction

Cloud computing has rapidly advanced, driving elastic scheduling and efficient utilization of computing resources. Among these, CPU utilization stands out as a core indicator, directly reflecting the operational state and scheduling efficiency of resources. With the widespread adoption of large-scale distributed data centers and virtualization technologies, the usage patterns of CPU resources exhibit complex dynamics and a high degree of uncertainty [1]. The concurrency of tasks, the diversity of applications, and the fluctuation of user requests make CPU utilization highly non-stationary and multi-scale over time. This complexity not only creates challenges for system resource management but also makes accurate CPU utilization prediction a key factor for intelligent scheduling and load balancing. Therefore, modeling and predicting CPU utilization time series holds both practical significance and theoretical value [2].

From the perspective of research background, CPU consumption in cloud computing environments shows multiple temporal dependency patterns. Short-term bursts of task requests may cause sudden fluctuations in CPU utilization, while long-running services may display periodic or trending characteristics. This multi-scale temporal structure often makes it difficult for traditional prediction methods to capture both global and local dynamics. In addition, coupling effects introduced by virtual machine migration, container orchestration, and multi-tenant parallel operations further increase the complexity and nonlinearity of CPU utilization fluctuations. Capturing these diverse patterns across multiple time scales has thus become a central challenge for predictive modeling [3].

In terms of significance, accurate CPU utilization prediction can enhance the intelligence of resource scheduling and directly improve service quality and energy efficiency in cloud computing platforms. When scheduling systems allocate resources based on precise predictions, they can avoid waste caused by over-provisioning and prevent bottlenecks or interruptions caused by insufficient resources. For large-scale platforms, this translates into higher reliability and lower operational costs [4]. At the same time, CPU utilization forecasting supports energy control and green computing by predicting high-load periods in advance, allowing for better resource management to reduce peak energy consumption and contribute to sustainable data center development.

Furthermore, research on CPU utilization prediction is essential for promoting the integration of cloud computing and artificial intelligence. In intelligent cloud platforms, scheduling and resource optimization are no longer based solely on static rules but are increasingly driven by data-oriented dynamic decision-making. High-quality time series forecasting models provide accurate prior information for such systems, enabling adaptive resource adjustment and fault prevention. This not only enhances resilience and stability but also lays a foundation for autonomous cloud platforms. In addition, insights gained from CPU utilization prediction can be transferred to other key

resources such as memory, storage, and network bandwidth, advancing intelligent resource management as a whole [5], [6].

In summary, conducting in-depth research on CPU utilization time series based on multi-scale modeling directly addresses the challenges of complexity and dynamism in cloud resource scheduling. It contributes to the enrichment of the theoretical framework for time series modeling and prediction, while also offering practical support for efficient operation and intelligent scheduling in data centers [7]. As cloud computing continues to expand and diversify in its applications, the importance of CPU utilization prediction will become even more prominent, providing indispensable support for sustainable growth and intelligent upgrades of cloud platforms.

Recent studies on cloud resource scheduling further clarify the operational value of CPU utilization forecasting. Reinforcement-learning-based task scheduling, collaborative scheduling for multi-tenant inference services, learning-based backend agents, and hierarchical communication-computation scheduling all treat resource allocation as a dynamic decision process driven by changing node states and workload pressure [8], [9], [10], [11]. These studies indicate that accurate utilization prediction should not be isolated from scheduling logic, as forecast outputs directly influence placement, scaling, and cross-node coordination. For this reason, multi-scale temporal modeling provides useful prior knowledge for intelligent cloud operations.

Research on backend anomaly detection and representation learning also supports the need for robust temporal feature extraction. Unified multi-granularity representations, structure-guided attention, uncertainty-aware failure detection, log semantic graphs, and Mamba-LSTM anomaly modeling emphasize that system behavior often contains hierarchical signals, delayed dependencies, and asymmetric risks [12], [13], [14], [15], [16]. Structure-aware Transformer modeling and retrieval-augmented long-text reasoning further show that multi-scale fusion and cross-context evidence integration can improve the stability of feature interpretation in complex environments [17], [18]. These ideas are consistent with the proposed multi-scale Transformer, which combines local fluctuation modeling with long-range dependency capture.

Broader intelligent risk and decision models provide complementary methodological references for cloud forecasting. Heterogeneous graph learning, temporal entity-relation modeling, unified multi-source feature aggregation, multi-agent decision making, dynamic interest evolution modeling, and graph-based advertising decision methods all demonstrate the importance of linking prediction with downstream risk identification and adaptive decision support [19], [20], [21], [22], [23], [24]. For CPU utilization forecasting, these studies suggest that prediction accuracy should ultimately serve more reliable resource scheduling, proactive risk control, and intelligent infrastructure management.

2. Proposed Approach

In this study, the core idea of the time series prediction method is to regard CPU utilization in a cloud computing environment as a multi-scale dynamic signal and capture its temporal dependencies and potential features at different levels through a multi-scale Transformer architecture. The overall model architecture is shown in Figure 1.

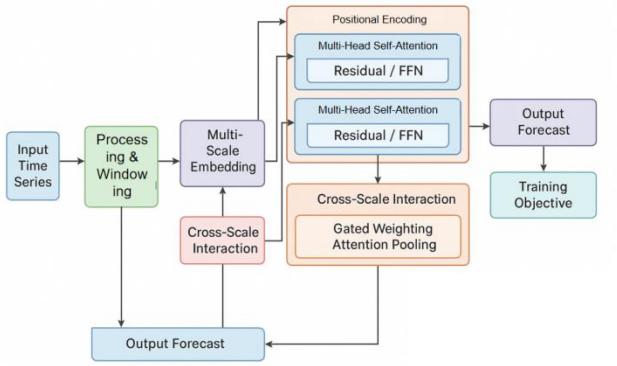

Figure 1. Overall model architecture

First, let the given time series be $X = \{x_1, x_2, ..., x_T\}$, where $x_t \in R$ represents the CPU utilization at time t. To enhance the characterization of features at different time granularities, the series is divided into subsequences $X^{(s)}$ of multiple scales and mapped into the feature space, respectively:

$$H^{(s)} = f_{embed}(X^{(s)}) \quad (1)$$

Where f_{embed} represents the embedding function and S is the number of multi-scale layers. In this way, the input is represented at short, medium, and long scales, providing multi-level feature support for subsequent modeling.

During the modeling phase, the key to the multi-scale Transformer lies in the multi-head self-attention mechanism. For the feature representation $H^{(s)}$ at scale s, its attention weight is calculated as follows:

$$Attention(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \quad (2)$$

Where $Q = H^{(s)}W_Q, K = H^{(s)}W_K, V = H^{(s)}W_V$ and W_Q, W_K, W_V are trainable parameters. Through the multi-head mechanism, the correlations of different subspaces can be modeled in parallel, thereby capturing both local patterns and long-range dependencies.

In the fusion stage, to make full use of information at different scales, the study adopts a cross-scale weighted fusion strategy. Specifically, assuming that the output of each scale is $\{Z^{(1)}, Z^{(2)}, ..., Z^{(S)}\}$, the final fusion is expressed as:

$$Z = \sum_{s=1}^{S} a_s Z^{(s)}, \sum_{s=1}^{S} a_s = 1 \quad (3)$$

Where a_s represents the importance weight of each scale, which is generated dynamically by learnable parameters or based on the attention mechanism. This fusion process ensures a balance between short-term fluctuations and long-term trends in a unified representation, thereby improving overall forecasting performance.

In the prediction phase, the fused sequence representation Z is input to the feedforward prediction module to generate future utilization estimates. Assuming the prediction step size is h, the predicted value is:

$$y_{t+h} = f_{pred}(Z_t) \tag{4}$$

Where f_{pred} is the prediction function composed of a multi-layer perceptron. To constrain the stability and prediction accuracy of the model, the mean square error (MSE) is used as the optimization target:

$$L = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{5}$$

Where y_i is the true value, $\hat{y}_i$ is the predicted value, and N is the number of samples. This optimization goal ensures that the model can accurately model and predict future CPU utilization based on multi-scale feature representation.

3. Experiment result

3.1 Dataset

The dataset used in this study is the Alibaba Cluster Trace 2017, which was publicly released by Alibaba Group on the Tianchi platform. It covers task scheduling and resource utilization in real-world internet data centers under production environments. The dataset contains operating logs from over 4, 000 servers, spans approximately eight days, and has a second-level sampling interval. It records information such as CPU usage, memory utilization, and task execution states. As a widely used public dataset in the field of cloud computing, it provides researchers with authentic and high-density traces of resource utilization.

In terms of data composition, the dataset not only includes CPU utilization as a key indicator but also provides multidimensional features such as job and task execution times, container allocation, resource requests, and actual usage. This information offers a comprehensive depiction of the dynamic changes in resource usage within cloud computing environments, supporting the modeling of both short-term workload bursts and long-term fluctuation trends. Particularly under virtualization and multi-tenant concurrency scenarios, CPU usage sequences exhibit clear non-stationarity and multi-scale characteristics, making the dataset highly suitable for validating the effectiveness of the proposed approach.

In the experimental design, the dataset was standardized and cleaned by removing missing values and outliers to ensure data quality. It was then partitioned sequentially into training, validation, and test sets in a 7:2:1 ratio, serving model training, parameter tuning, and final performance evaluation. As a publicly available resource, this dataset guarantees the reproducibility and verifiability of the experimental process and results, while also facilitating further research and comparative analysis within the academic community.

3.2 Experimental Results

This paper first conducts a comparative experiment, and the experimental results are shown in Table 1.

Table 1. Comparative experimental results

Method	MSE	R^2	MAE	RMSE
LSTM [25]	0.0231	0.871	0.107	0.152
Transformer [26]	0.0194	0.896	0.096	0.139
1DCNN [27]	0.0257	0.854	0.113	0.160
BILSTM [28]	0.0217	0.884	0.102	0.147
Ours	0.0172	0.911	0.090	0.131

From Table 1, it can be observed that traditional recurrent neural network models such as LSTM and BILSTM can capture certain temporal dependencies when predicting CPU utilization time series. They therefore show relatively good fitting ability. However, due to inherent limitations in modeling long-range dependencies, their performance is not optimal in cloud computing scenarios with multi-scale fluctuations and complex dynamic features. In particular, the higher MSE and RMSE values of LSTM indicate its inadequacy in representing multi-scale variations.

In comparison, the Transformer model outperforms LSTM and BILSTM in most metrics. This shows that the self-attention mechanism is more effective in modeling global dependencies and capturing dynamic features across multiple scales. In cloud computing environments, CPU utilization is often influenced by both short-term bursty requests and long-term load trends. The multi-head self-attention mechanism enables effective modeling across different time spans, which significantly improves prediction accuracy. Its R^2 value is close to 0.90, which demonstrates strong explanatory power.

By contrast, the prediction performance of 1DCNN is the weakest. This result aligns with the limitations of its feature extraction approach. Although convolutional structures are effective in local pattern recognition, relying only on local receptive fields is insufficient to capture long-term dependencies and cross-scale dynamics in complex time series. Consequently, it performs poorly in all error metrics. This further verifies that in CPU utilization prediction for cloud computing, shallow or purely local modeling is inadequate for handling diverse temporal features.

The proposed method achieves the best performance across all metrics. In particular, MSE and RMSE are significantly reduced compared with other models, while R^2 increases to 0.911. These results demonstrate the effectiveness of the multi-scale Transformer architecture in CPU utilization prediction. The introduction of multi-scale embeddings and cross-scale interaction enables comprehensive modeling of multiple temporal dependencies in cloud resource usage. This leads to higher prediction accuracy and provides strong support for intelligent scheduling and resource optimization.

This paper also gives the experimental results of the sensitivity of batch size to RMSE, and the experimental results are shown in Figure 2.

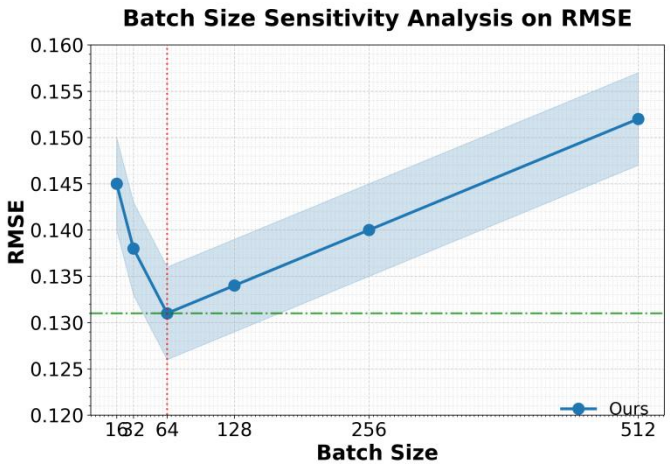

Figure 2. Sensitivity of Batch Size to RMSE

From Figure 2, it can be seen that batch size has a significant impact on RMSE. With smaller batch sizes such as 16 and 32, the RMSE is higher. This indicates that the model is more affected by gradient fluctuations and unstable updates under such training settings, leading to reduced prediction accuracy. When the batch size increases to 64, the RMSE reaches its lowest value, suggesting that the model achieves a good balance between stability and generalization at this point.

As the batch size continues to increase to 128 and beyond, RMSE shows a gradual upward trend. This suggests that overly large batch sizes may weaken the model's ability to capture local fluctuation features. In particular, within the multi-scale fluctuations of CPU utilization in cloud computing, large batch updates make the gradients too smooth. This reduces the model's sensitivity to short-term burst changes and results in lower prediction accuracy.

Overall, the optimal RMSE is observed when the batch size is around 64. This highlights the critical role of selecting an appropriate batch size in training under the multi-scale Transformer architecture. A suitable choice not only reduces instability caused by gradient noise but also ensures effective learning under complex temporal patterns. For cloud computing scenarios, this means that careful tuning of training hyperparameters can better support CPU utilization prediction tasks.

This experimental result underlines the importance of hyperparameter sensitivity analysis. The batch size sensitivity test guides future training and deployment of the model. It helps avoid energy waste or performance bottlenecks caused by inaccurate predictions in resource scheduling. The results also confirm the advantages of the multi-scale Transformer in CPU utilization modeling for cloud computing, while emphasizing its dependence on hyperparameter configuration, which requires careful adjustment in practice.

This paper also gives the sensitivity of multi-scale quantities to R^2, and the experimental results are shown in Figure 3.

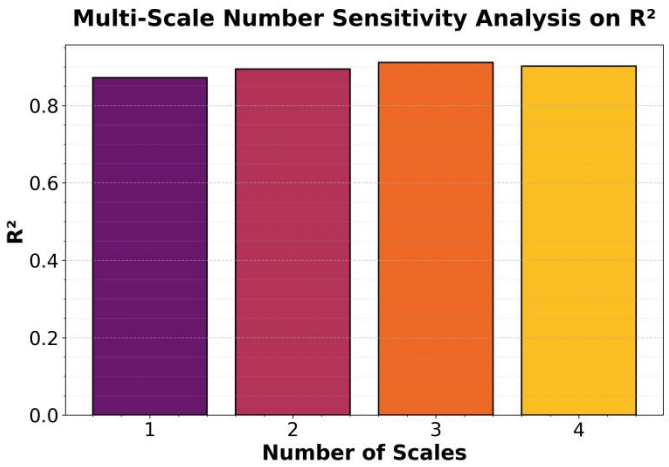

Figure 3. Sensitivity of multi-scale quantity to R^2

From Figure 3, it can be observed that the variation in the number of scales has a certain sensitivity to the R^2 metric. When the number of scales is 1, the R^2 value is relatively low. This indicates that a single-scale modeling approach cannot fully capture the complex patterns in CPU utilization time series. In a cloud computing environment, single-scale features are insufficient to represent both short-term bursts and long-term trends, which limits prediction accuracy.

When the number of scales increases to 2 and 3, the R^2 value gradually improves and reaches its highest point at 3 scales. This shows that the multi-scale mechanism effectively enhances the model's feature representation ability. It allows the extraction of valuable information at different temporal granularities. In particular, in CPU utilization prediction for cloud computing, the three-scale configuration enables the model to balance short-term dependencies and global trends, which significantly improves its ability to explain complex temporal dynamics.

However, when the number of scales further increases to 4, the R^2 value decreases slightly. This result indicates that too many scales may introduce redundant features or noisy information, which reduces the model's predictive ability. This finding suggests that multi-scale modeling requires a balance between sufficient information and avoiding redundancy. Otherwise, the model may face problems of overfitting or interference from ineffective features.

In summary, the multi-scale mechanism plays a significant role in improving CPU utilization prediction in cloud computing, but its effect lies within an optimal range. An appropriate number of scales helps the model fully capture temporal features and improve prediction accuracy. Too few or too many scales can lead to performance decline. Therefore, determining the proper number of scales is not only a key step in model design but also an essential condition for maintaining stability and high accuracy in complex cloud computing environments.

4. Conclusion

This study focuses on the problem of CPU utilization time series prediction in cloud computing environments and proposes a modeling framework based on a multi-scale Transformer. The method captures complex temporal

dependencies through multi-scale embeddings and cross-scale interaction. It balances short-term bursts and long-term trends. In the overall design, the self-attention mechanism provides the ability to model global dependencies, while the multi-scale structure enhances feature representation at different temporal granularities. As a result, the model shows superior predictive performance in dynamic and complex cloud computing environments.

In the experimental evaluation, the proposed model achieved better performance than traditional methods on multiple error and fitting metrics. This confirms the effectiveness of the multi-scale Transformer in CPU utilization prediction. Comparisons with recurrent neural networks, convolutional neural networks, and the standard Transformer show clear advantages in both stability and accuracy. The results indicate that integrating multi-scale features into the attention mechanism not only alleviates the challenge of long-term dependency but also improves robustness when facing complex fluctuating data.

From an application perspective, the proposed method has direct value for cloud computing resource scheduling and management. Accurate CPU utilization prediction enables data centers to make more rational decisions in resource allocation, load balancing, and energy consumption control. By integrating information across different time scales, the prediction model provides high-quality prior knowledge for the system. This reduces the risk of service interruption, lowers energy waste, and improves overall operational efficiency. These benefits are of great practical significance for large-scale distributed computing platforms.

In conclusion, this study not only extends the theoretical system of time series prediction at the academic level but also provides a feasible path for resource management in engineering practice. By exploring CPU utilization prediction in depth, the method demonstrates strong potential in multi-scale dynamic modeling and promotes the application of time series prediction in cloud computing. Its value lies not only in improving prediction accuracy but also in enhancing system stability, energy efficiency, and intelligent scheduling, which lays a solid foundation for the development of related fields.

References

[1] G. Cui, T. Hu, W. Zhang, et al., "MMTransformer: a multivariate time-series resource forecasting model for multi-component applications," Scientific Reports, vol. 15, no. 1, p. 29740, 2025.

[2] S. Liu, K. Xiong, Y. Li, et al., "SDVformer: A Resource Prediction Method for Cloud Computing Systems," Computers, Materials & Continua, vol. 84, no. 3, 2025.

[3] M. Smendowski and P. Nawrocki, "Optimizing multi-time series forecasting for enhanced cloud resource utilization based on machine learning," Knowledge-Based Systems, vol. 304, p. 112489, 2024.

[4] M. Li, X. Zhang, Y. Liu, et al., "A Multiscale Transformer Model for Long Time Series Forecasting Based on Discrete Wavelet Transform and Residual Learning Modules," Journal of Forecasting, 2025.

[5] W. Zheng, W. Zheng, Y. Chen, et al., "Wavelet-enhanced Dual Branch Transformer for Accurate Cloud Workload Prediction," in Proceedings of the 2025 International Conference on Artificial Intelligence and Computational Intelligence, 2025, pp. 459-464.

[6] N. Persson Suorra, "Multivariate Resource Usage Forecasting and Temporal Accuracy in Private Cloud Systems," 2025.

[7] P. Chen, H. Ye, F. Jiang, et al., "Multi-scale transformers with adaptive pathways for time series forecasting," 2024.

[8] R. Wei and D. Qiao, "Task Scheduling Research for Cloud Computing Infrastructures: A Reinforcement Learning Method Integrating Resource Occupancy Prediction, Dynamic Node State Encoding, and Placement Profit Maximization," 2025.

[9] Z. Huang, J. Luo, and C. Chen, "Learning-Driven Collaborative Scheduling for Multi-Tenant Distributed Inference Services," 2025.

[10] L. Ren, Y. Liu, and Y. Zhao, "Learning-Based Intelligent Agents for Backend Resource Scheduling and Operational Decision Making," 2025.

[11] R. Meng, "Hierarchical Communication and Computation Scheduling for Efficient Federated Learning in Cloud-Edge Collaborative Intelligent Systems," 2024.

[12] Y. Yang, C. Wen, H. Cui, Y. Sun, and Y. Liu, "Learning Unified Multi-Granularity Representations for Backend Anomaly Detection and Causal Localization," 2024.

[13] X. Li, R. Yin, and S. Dang, "Structure-Guided Attention and Multi-Scale Behavior Modeling for Microservice Anomaly Detection," 2025.

[14] Y. Liu, Z. Zhang, and S. Liang, "Intelligent Backend Failure Detection with Uncertainty Quantification and Asymmetric Risk Optimization," 2024.

[15] Y. Yang and A. Xu, "Log Semantic Graph Construction and System Event Anomaly Detection via Convolutional Neural Networks," 2024.

[16] Y. Zhang and E. Cheng, "Anomaly Detection in E-Commerce Multi-Table ETL Processes through Mamba-LSTM Collaborative Modeling," 2025.

[17] K. Zeng, "Transformer-Based Structure-Aware Lung Cancer Image Segmentation with Multi-Scale Fusion and Boundary-Guided Prediction," 2024.

[18] S. Y. Huang, "Retrieval-Augmented Long-Text Reasoning with Semantic Conflict Detection and Cross-Paragraph Evidence Integration," 2024.

[19] Y. Xu, "Intelligent Supply Chain Risk Identification via Heterogeneous Graph Learning and Multi-Entity Interaction Modeling," 2025.

[20] R. Yan and M. Guo, "Enterprise Audit Risk Identification Through Temporal Context and Entity Relationship Anomaly Modeling," 2024.

[21] Y. Nie, "Financial Risk Identification Using Unified Multi-Source Feature Learning and Structural Aggregation," 2024.

[22] S. Sun, "Adaptive Stock Trading Algorithms Jointly Driven by Multi-Agent Collaborative Decision Making and Large Language Models in Complex Financial Market Environments," 2025.

[23] Z. Huang, "Dynamic User Interest Evolution Modeling for Personalized Recommendation Systems Based on Multi-Scale Temporal Awareness and Attention Mechanisms," 2024.

[24] Z. Pan, "Intelligent Decision-Making for Advertising Recommendation via Data Mining and Graph Neural Networks," 2024.

[25] I. A. Abbasi, "Adapted convolutional neural networks and long short-term memory for host utilization prediction in cloud data center," 2021.

[26] J. Chen, H. Ye, F. Jiang, et al., "Fremer: Lightweight and Effective Frequency Transformer for Workload Forecasting in Cloud Services," arXiv preprint arXiv:2507.12908, 2025.

[27] A. Ullah, S. F. A. Razak, S. Yogarayan, et al., "Modified Neural Network Used for Host Utilization Predication in Cloud Computing Environment," Computers, Materials & Continua, vol. 82, no. 3, 2025.

[28] J. Han and P. Zeng, "Residual BiLSTM based hybrid model for short-term load forecasting in buildings," 2025.