Proceedings of Machine Learning Research vol 278:1–6, 2025 2025 2nd International Conference on Machine Learning and Intelligent Computing

Cloud Resource Auto-Scaling Strategy Based on CNN-Lightweight Transformer

Yue Zhang ZHANGYUE@SIA.CN
State Key Laboratory of Robotics
Shenyang Institute of Automation, Chinese Academy of Sciences
Shenyang 110016, China

Chunhe Song[*] SONGCHUNHE@SIA.CN
Key Laboratory of Networked Control Systems
Chinese Academy of Sciences
Shenyang 110016, China
** Corresponding author*

Editors: Nianyin Zeng, Ram Bilas Pachori and Dongshu Wang

Abstract

With the rapid development of cloud computing and containerization technologies, load forecasting has become increasingly important in resource management. This paper proposes a load forecasting model based on a lightweight Transformer and local convolution fusion, aiming to efficiently capture multi-scale features of complex loads while maintaining low computational overhead. Furthermore, this paper introduces a predictive error feedback and adaptive cooling period adjustment mechanism based on traditional Horizontal Pod Autoscaling (HPA), enhancing the system's adaptability to load variations by dynamically adjusting scaling strategies. Experimental results demonstrate that the proposed model excels in both load forecasting accuracy and scheduling stability, effectively balancing response speed and system robustness, providing an efficient solution for cloud resource management.

Keywords: Load Forecasting; Transformer; CNN; Resource Management; Auto-Scaling

1. Introduction

As cloud computing technology advances, more enterprises are migrating to cloud platforms for flexibility and scalability (Darwish, 2024). However, the volatility and complexity of workloads in cloud environments pose challenges for resource management. Effective load forecasting can help cloud service providers optimize resource allocation, reduce costs, improve user experience, and ensure service availability, making it an important research direction in cloud computing.

Traditional load forecasting methods rely on time series analysis and statistical models like ARIMA and Exponential Smoothing, which work well with linear data but struggle with complex nonlinear patterns (Smyl et al., 2024). Recently, deep learning technologies, particularly LSTM and CNN, have gained attention for their ability to capture complex patterns (Eren and Küçükdemiral, 2024).

Despite their success, deep learning models face challenges such as data scarcity and overfitting. Additionally, traditional scheduling mechanisms lack adaptability to dynamic load changes, leading to resource waste. Designing efficient and flexible load forecasting models and optimizing scheduling mechanisms have become research hotspots.

This paper proposes a load forecasting model based on a lightweight Transformer and local convolution fusion, combined with predictive error feedback and adaptive cooling period adjustment mechanisms, to improve forecasting accuracy and scheduling stability, aiming to provide an efficient solution for cloud resource management.

2. Related Work

In recent years, load forecasting has gained significant attention in cloud computing resource management. With the rapid development of cloud computing and containerization technologies, effectively managing and scheduling resources has become a key challenge in this field. The volatility and complexity of workloads in cloud environments make dynamic resource allocation and optimization particularly important, especially in response to changing user demands and service quality requirements. Existing research primarily employs time series analysis methods, such as ARIMA and Exponential Smoothing, which perform well with linear data but struggle with complex nonlinear load patterns. Recently, deep learning models, such as LSTM and CNN, have gained attention for their ability to capture complex patterns, but they may face training difficulties and overfitting issues when data is scarce or computational resources are limited. Additionally, reinforcement learning is gradually being applied to load forecasting and resource scheduling, allowing intelligent agents to adjust resource allocation in real time; however, this approach requires complex training and substantial interaction data (Zhou et al., 2024). Despite significant advancements in load forecasting, challenges remain in addressing data scarcity and adaptability to dynamic load changes (Tsoumplekas et al., 2025). To tackle these issues, this paper proposes a novel load forecasting model based on a lightweight Transformer and local convolution fusion, aiming to improve forecasting accuracy and scheduling stability (L'Heureux et al., 2022). Compared to existing methods, the proposed model has unique advantages in handling complex load patterns, effectively capturing multi-scale features while maintaining low computational overhead. Furthermore, the scheduling strategy that incorporates an adaptive cooling period adjustment mechanism can respond more flexibly to load changes, enhancing resource utilization and reducing system volatility.

3. Research Content

3.1. Model Architecture Design

The load forecasting model proposed in this study combines a lightweight Transformer (Ekambaram et al., 2023) and local convolution to improve prediction accuracy and efficiency. As shown in Figure 1, the architecture features a convolutional embedding layer, a lightweight Transformer encoder, residual connections with layer normalization, and a prediction output layer. The convolutional layer uses 1D convolution to extract local temporal features, reducing dimensionality and noise. The lightweight Transformer encoder captures global dependencies with fewer self-attention layers, balancing short-term fluctuations and long-term trends while enhancing training speed. Residual connections and layer normalization stabilize information transfer and address the vanishing gradient problem. The prediction output layer generates load forecasts for future periods, supporting both short-term and multi-step predictions, thus enhancing the model's flexibility and adaptability across various load scenarios (Jiang et al., 2024).

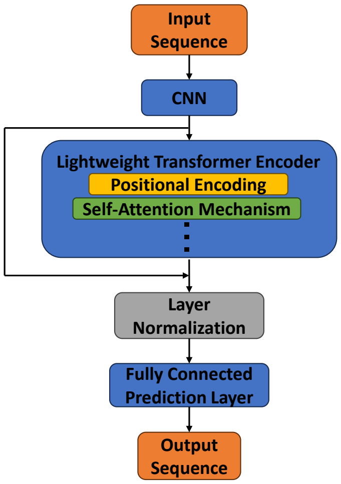

Figure 1: Model Architecture.

3.2. Innovation in Scheduling Mechanism

This study introduces a predictive error feedback and adaptive cooling period adjustment module for scheduling, based on traditional Horizontal Pod Autoscaling (HPA) (Van Do et al., 2025). As shown in Figure 2, this mechanism dynamically adjusts scaling strategies by calculating real-time prediction errors between historical load data and the forecasting model. The prediction error threshold is determined experimentally, and the cooling period is dynamically adjusted based on the error size: when the prediction error is small, the system quickly adopts an aggressive scaling strategy for timely resource allocation; conversely, with a large prediction error, it extends the cooling period or scales based on current request counts, using a conservative strategy to avoid frequent scheduling errors. This adaptive mechanism enhances the system's adaptability to load changes, optimizes resource utilization, and reduces volatility, ultimately improving resource management efficiency and stability in cloud computing environments (Zhou et al., 2023).

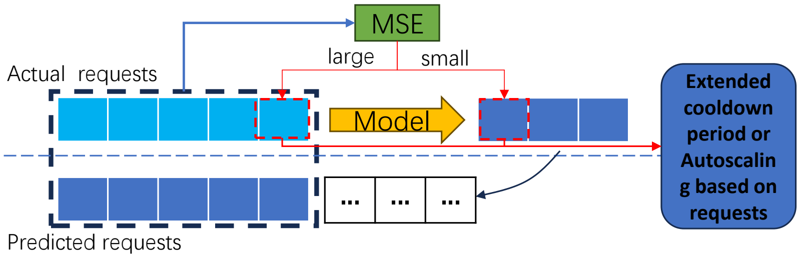

Figure 2: Scheduling Mechanism.

4. Experiments

This study utilizes the FIFA dataset, which records HTTP request logs during the 1998 World Cup. This dataset effectively simulates specific load patterns and is suitable for predicting request volumes because the number of requests surged dramatically during the matches, simulating traffic fluctuations caused by large-scale events.

The experiments were conducted on a Linux system equipped with an NVIDIA GeForce RTX 4090 GPU, and the parameter settings are shown in the Table 1.

Table 1: Parameter Settings.

Training Parameters		**CNN Parameters**		**Transformer Parameters**	
Learning Rate	0.001	Number of Layers	1	D Model	64
Batch Size	32	Input Channels	16	Number of Layers	1
Epochs	10	Output Channels	16	Feed Forward Dimension	128
Optimizer	Adam	Kernel Size	3	Dropout Rate	0.1

In this study, the design of the model architecture considers multiple factors. The lightweight transformer is set to 1 layer based on its effectiveness in handling sequential data. The CNN kernel size is set to 3, which has been validated through experiments to effectively capture features in the input data. The parameters we selected were adjusted through multiple experiments to ensure the model achieves optimal performance in predicting request volumes.

During data cleaning, we check for missing values and decide whether to fill or delete them, while also removing duplicate records to ensure data uniqueness. For outliers, we identify and handle them using statistical methods, choosing to delete or replace them to minimize their impact on model training.This study employs regularization and hyperparameter optimization techniques during model training to ensure the model's generalization ability and the integrity of the training process.

The evaluation metrics are divided into two categories: Model accuracy metrics (such as Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE)) are used to quantify prediction accuracy; Scheduling algorithm effectiveness metrics (such as resource shortage indicator (θ_U), resource surplus indicator (θ_O), resource shortage time (T_U), resource surplus time (T_O), and elasticity gain improvement indicator (ϵ_n)) are used to evaluate resource management efficiency and system responsiveness. These metrics measure whether the current number of Pods meets the optimal number, the proportion of time the system is in a state of insufficient or excessive resource supply, and the efficiency comparison between autoscaling and non-autoscaling. These metrics provide a basis for validating the effectiveness of the load forecasting model and scheduling algorithm (Senjab et al., 2023).

5. Analysis of Experimental Results

5.1. Analysis of Model Prediction Accuracy

Through experiments on the FIFA dataset, this study validates the effectiveness of the load forecasting model based on a lightweight Transformer and local convolution. The prediction errors of different models are presented in Table 2. The results demonstrate that the proposed model excels in load forecasting, accurately capturing fluctuations, particularly during peak traffic periods. Its prediction accuracy metrics, including Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE), surpass those of other models, indicating high accuracy

Table 2: Prediction errors of different models.

Model	Prediction Error		
	RMSE	MAE	MAPE
LSTM	5937.2	3691.3	107.5
GRU	4863.2	4263.8	96.4
CNN-Transformer	1535.4	1308.9	75.1

Table 3: Comparison of evaluation indicators for different algorithms.

Evaluation indicators	No autoscaling	HPA	CNN-Transformer
θ_U	1.96	7.64	0.83
θ_O	213.6	10.37	85.47
T_U	19.7	67.42	6.15
T_O	186.3	45.87	97.68
ε_n	1	1.58	2.45

even during sudden traffic surges. Comparative experiments further confirm the model's superiority across various load patterns, highlighting that the combination of a lightweight Transformer and local convolution significantly enhances forecasting accuracy.

5.2. Analysis of the Scheduling Algorithm

In evaluating the scheduling mechanism, the adaptive cooling period adjustment module based on prediction results significantly enhances resource management efficiency. By calculating the real-time prediction error between historical load data and the forecasting model, the system dynamically adjusts scaling strategies. When the prediction error is small, it employs an aggressive scaling strategy for timely resource allocation. Conversely, with a large prediction error, the system extends the cooling period or scales based on the current request count, adopting a conservative strategy to minimize frequent scheduling due to inaccurate predictions. The comparison of evaluation indicators for different algorithms is shown in Table 3. Experimental results indicate a significant reduction in the resource shortage indicator (θ_U), demonstrating the algorithm's effectiveness in minimizing shortages. However, there is still potential for improvement in controlling resource surplus (θ_O), while the expansion gain is notably enhanced. Overall, the scheduling mechanism successfully improves resource utilization efficiency and reduces system volatility, validating the proposed strategy's effectiveness in practical applications.

6. Conclusion

With the rapid growth of cloud computing, load forecasting is essential for resource management. This paper proposes a model combining a lightweight Transformer and local convolution to effectively capture complex workload features while reducing computational overhead. For scheduling, a predictive error feedback and adaptive cooling period adjustment module based on HPA dynamically adjusts scaling strategies, improving adaptability and reducing scheduling frequency. Experiments show that the model achieves high accuracy and stability in load forecasting, offering an efficient solution for cloud resource management (Khan et al., 2022).

References

Dina Darwish. Emerging trends in cloud computing analytics, scalability, and service models. 2024.

Vijay Ekambaram, Arindam Jati, Nam Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. Tsmixer: Lightweight mlp-mixer model for multivariate time series forecasting. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*, pages 459–469, 2023.

Yavuz Eren and İbrahim Küçükdemiral. A comprehensive review on deep learning approaches for short-term load forecasting. *Renewable and Sustainable Energy Reviews*, 189:114031, 2024.

Bozhen Jiang, Hongyuan Yang, Yidi Wang, Yi Liu, Hua Geng, Huarong Zeng, and Jiangqiao Ding. Dynamic temporal dependency model for multiple steps ahead short-term load forecasting of power system. *IEEE Transactions on Industry Applications*, 2024.

Tahseen Khan, Wenhong Tian, Guangyao Zhou, Shashikant Ilager, Mingming Gong, and Rajkumar Buyya. Machine learning (ml)-centric resource management in cloud computing: A review and future directions. *Journal of Network and Computer Applications*, 204:103405, 2022.

Alexandra L'Heureux, Katarina Grolinger, and Miriam AM Capretz. Transformer-based model for electrical load forecasting. *Energies*, 15(14):4993, 2022.

Khaldoun Senjab, Sohail Abbas, Naveed Ahmed, and Atta ur Rehman Khan. A survey of kubernetes scheduling algorithms. *Journal of Cloud Computing*, 12(1):87, 2023.

Slawek Smyl, Grzegorz Dudek, and Paweł Pełka. Contextually enhanced es-drnn with dynamic attention for short-term load forecasting. *Neural Networks*, 169:660–672, 2024.

Georgios Tsoumplekas, Christos Athanasiadis, Dimitrios I Doukas, Antonios Chrysopoulos, and Pericles Mitkas. Few-shot load forecasting under data scarcity in smart grids: A meta-learning approach. *Energies*, 18(3):742, 2025.

Tien Van Do, Nam H Do, Csaba Rotter, TV Lakshman, Csaba Biro, and T Bérczes. Properties of horizontal pod autoscaling algorithms and application for scaling cloud-native network functions. *IEEE Transactions on Network and Service Management*, 2025.

Guangyao Zhou, Wenhong Tian, Rajkumar Buyya, Ruini Xue, and Liang Song. Deep reinforcement learning-based methods for resource scheduling in cloud computing: A review and future directions. *Artificial Intelligence Review*, 57(5):124, 2024.

Zhiqiang Zhou, Chaoli Zhang, Lingna Ma, Jing Gu, Huajie Qian, Qingsong Wen, Liang Sun, Peng Li, and Zhimin Tang. Ahpa: adaptive horizontal pod autoscaling systems on alibaba cloud container service for kubernetes. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 15621–15629, 2023.