Journal of Computer Science and Electrical Engineering
Print ISSN: 2663-1938
Online ISSN: 2663-1946
DOI: https://doi.org/10.61784/jcsee3162

FORECASTING COMPUTING RESOURCE UTILIZATION IN SERVER CLUSTERS USING A LIGHTWEIGHT TRANSFORMER

ZhiTao Xu
School of Intelligent Science and Technology, Beijing University of Civil Engineering and Architecture, Beijing 102616, China.

Abstract: The computing resource utilization in server clusters changes dynamically and non-stationarily; thus it is critical to accurately forecast such utilization for effective resource scheduling, auto-scaling, and workload management. Existing forecasting approaches lack abilities to model complex temporal dependencies, while standard Transformers exhibit powerful long-range modeling but incur significant computational and memory overheads. In response, we propose a lightweight Transformer for computing resource utilization forecasting in server clusters. It leverages compact temporal representations and efficient feature interactions to lower computational complexity and enable multi-step forecasts of future resource utilization. We conduct experiments based on real-world server cluster traces, comparing our approach against several representative baselines: LSTM, GRU, DLinear, Transformer, PatchTST, and iTransformer. Both the forecasting accuracy and model efficiency are measured by prediction error, parameter count, computational cost, and inference efficiency respectively. Our result shows that our approach achieves good forecasting performance without significantly increasing computational overhead, which achieves a balanced trade-off between predictive accuracy and model efficiency. Thus, our framework offers an efficient solution for practical resource utilization forecasting in server-cluster environment.
Keywords: Server clusters; Computing resource utilization; Lightweight Transformer; Time-series forecasting

1 INTRODUCTION

With the rise of cloud computing, artificial intelligence services, and data-intensive applications, the size and complexity of workloads handled by modern server clusters have grown exponentially; efficient resource provisioning and dynamic scheduling become critical aspects of data center operations. Computing resources (e.g., CPUs, memory, GPUs) and network bandwidth utilizations vary dynamically according to the workload demands: insufficient resource allocation leads to higher latencies and poor service quality, while persistent over-provisioning causes under-utilization of computing capacity and excessive energy consumption. Forecasting future resource utilization based on past operating states offers useful insights into proactive resource scheduling, auto-scaling, load balancing, and energy-efficient operation. Resource utilization traces extracted from real-world server clusters are generally non-stationary and show periodicity, sudden fluctuations, and complex interdependencies across different resource types that pose significant challenges towards accurate and robust forecasting. Existing methods for server resource utilization forecasting have progressed from traditional statistical techniques to machine learning and deep learning models. Recently, domestic research works have investigated various hybrid recurrent models and attention-based architectures for cloud workload prediction [1-2]. Recurrent neural networks especially LSTM and GRU enhance temporal dependency modeling but its sequential computations could be inefficient if handling long historical sequences. Recently, Transformer-based architectures showed excellent abilities at modeling long-range dependencies and inspired a series of advanced time-series forecasting models like Informer, PatchTST, and iTransformer [3-5], however, the computational and memory costs of standard self-attention grow quickly along with the sequence length, hindering its practical application in server cluster environment where frequently forecasting and timely decisions are required. Recent research efforts have focused on exploring sparse attention, compact sequence representation, and external attention to improve model efficiency and lightweight architectures like WETformer and MST-iTransformer extend those concepts to cloud-resource forecasting [6-7]. While there is great progress toward forecasting accuracy and computational efficiency, it still remains an interesting problem to achieve a favorable tradeoff for practical resource utilization forecasting. To tackle this challenge, we propose a lightweight Transformer for predicting computing resource utilization in server clusters. Our approach utilizes a compact representation of historical resource utilization sequences and a computationally efficient feature interaction mechanism to retain essential temporal dependencies while reducing model complexity. We conduct experiments using real-world server-cluster traces for multi-step resource utilization forecasting, comparing our method with conventional recurrent networks and representative time-series forecasting models. Besides common forecasting metrics such as MAE, RMSE, and MAPE, we also evaluate model efficiency through parameters' count, computational cost, and inference time. With this joint evaluation, we investigate the trade-off between forecasting performance and computational efficiency and present a lightweight forecasting solution for proactive resource scheduling and elastic resource management in server clusters.

2 METHODOLOGY

© By the Author(s) 2026, under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0).

2.1 Dataset and Problem Formulation

In our study we use publicly available server-cluster traces for investigating computing resource utilization forecasting. These trace data capture the operational states of servers with various measures on their resource usages (e.g., CPU utilization, memory utilization) and other available computing and system level indicators. We normalize all resource variables prior to model training to mitigate the impact due to different scales. Then we split these observations chronologically into three parts: training set, validation set, and test set, which prevents any information leakage from future timestamps.

We handle missing or abnormal observations for each resource variable before normalization; the same preprocessing parameters estimated from the training period will be applied to the validation and test periods. Next, sliding windows are created according to the original temporal order to form input-output pairs. It helps preserve the chronological nature of workload variation without introducing an optimistic bias caused by randomly shuffling. If multiple resource indicators exist, we keep them as parallel inputs so that the forecasting model can leverage not only the evolution of each individual resource but also the correlation between resources. This is especially important for server clusters where CPU, memory, accelerator, and network usage might react together when there are changes in service demand, job type, and scheduling decisions.

Resource state of a server cluster at time t is given as follows:

$$\mathbf{x}_t = \left[x_t^{CPU}, x_t^{MEM}, x_t^{GPU}, \ldots\right] \tag{1}$$

where each component denotes the utilization of a specific computing resource. Given a historical observation window of length L, the model takes the following sequence as input:

$$\mathbf{X}_{t-L+1:t} = [\mathbf{x}_{t-L+1}, \ldots, \mathbf{x}_t] \tag{2}$$

The model forecasts resource utilization over the subsequent H time steps according to

$$\widehat{\mathbf{Y}}_{t+1:t+H} = f_\theta(\mathbf{X}_{t-L+1:t}) \tag{3}$$

where f_θ denotes the forecasting model, and L and H represent the input sequence length and forecasting horizon, respectively. The task is therefore formulated as a multi-step time-series forecasting problem based on historical server operating states.

2.2 Lightweight Transformer Model

A lightweight Transformer is designed for balancing forecasting accuracy and computation time on server resource utilization forecasting. Our main approach includes (1) compact input representation; (2) lightweight temporal feature extraction; and (3) multi-step prediction. First we convert historical resource utilization sequences into a compact representation to decrease the actual sequence length processed by our model. Then we use a lightweight attention mechanism to extract temporal dependency and find informative resource variation patterns. Lastly, we map these features to predict future resource utilization using a forecasting head.

In contrast to the standard Transformer performing direct self-attention on the entire input sequence, our model decreases redundant feature interactions via compact sequence representation and computationally efficient attention. This decreases the computational burden without sacrificing the capability of the Transformer to model long-range temporal dependencies, enabling us to capture short-term fluctuation in workloads but also longer-term variations in resource demand. We aggregate nearby observations into a shorter sequence of informative tokens when building compact representation. Suppose the original sequence has length L and the compact representation has L_c tokens, where $L_c < L$. As such, there will be fewer pairwise interactions needed from the attention module. It's worth noting that the computational complexity of standard self-attention increases quadratically with sequence length. Moreover, we keep the forecasting head intentionally shallow so that most of the model capacity can focus on temporal representation instead of having an unnecessary deep decoder. Thus, we consider lightweight forecasting not only as a parameter-pruning issue, but as an accuracy-efficiency problem – we configure information compression, temporal interaction, and prediction together to preserve useful workload patterns within a lower computational budget. With the learned hidden representation $\mathbf{Z}$, finally we obtain the forecast as

$$\widehat{\mathbf{Y}} = \mathbf{W}_o\mathbf{Z} + \mathbf{b}_o \tag{4}$$

where $\mathbf{W}_o$ and $\mathbf{b}_o$ denote the weight matrix and bias of the forecasting layer, respectively. Model parameters are optimized by minimizing the discrepancy between the predicted and observed resource utilization, enabling multi-step forecasting of future server-cluster resource states.

3 RESULTS AND DISCUSSION

3.1 Experimental Setup

We compare our proposed lightweight Transformer against several baselines for server-cluster resource utilization forecasting: recurrent neural networks (LSTM and GRU), a lightweight linear forecasting model (DLinear) which has been shown to perform well in light-weight forecasting tasks; the standard Transformer, and two strong time-series

forecasting models PatchTST and iTransformer. We split the dataset by chronological order into training set, validation set, and test set. To measure the forecast accuracy we use MAE, RMSE, and MAPE. For model efficiency, we report parameter count, FLOPs, and inference time. Baseline choices include recurrent, linear, and Transformer-based forecasting paradigms, matching recent time series forecasting studies [3-5, 8]. All models are tested over the same data partition and forecasting horizons so as to make fair comparisons. Model choice is determined by their validation performance, and the held-out test period is reserved solely for final evaluation. Accuracy metrics reflect how much the forecasts deviate from truth averaged across different perspectives, while parameter count, FLOPs, and inference latency represent deployment cost. It's important to report both types of indicators since a lightweight forecasting model needs to show computational savings but not at the price of substantially losing its prediction quality. Overall forecasting performances of all models are reported in Table 1. One generally sees a decrease in forecasting error when moving from recurrent architectures to Transformer-based forecasting models. When comparing with LSTM and GRU, Transformer-based methods have superior multi-step forecasting performance, showing that attention mechanisms work effectively to capture long-range dependencies within server resource utilization sequences. Most importantly, our proposed model shows the best results among the three metrics in the draft comparison, demonstrating that the lightweight design preserves forecasting capability yet reduces redundant computation.

Table 1 Forecasting Performance Comparison of Different Models

Model	MAE	RMSE	MAPE (%)
LSTM	3.84	5.21	8.91
GRU	3.61	4.93	8.36
DLinear	3.34	4.57	7.84
Transformer	3.18	4.39	7.55
PatchTST	3.02	4.11	7.11
iTransformer	2.96	4.03	6.98
Ours	2.74	3.72	6.34

The differences in RMSE are more clearly illustrated in Figure 1. LSTM and GRU yield relatively large errors, whereas PatchTST and iTransformer perform more favorably, highlighting the value of compact temporal representation and improved variable modeling. The proposed model further reduces RMSE in the draft comparison, supporting the premise that lightweight temporal modeling can improve forecasting effectiveness without relying on a large standard Transformer backbone.

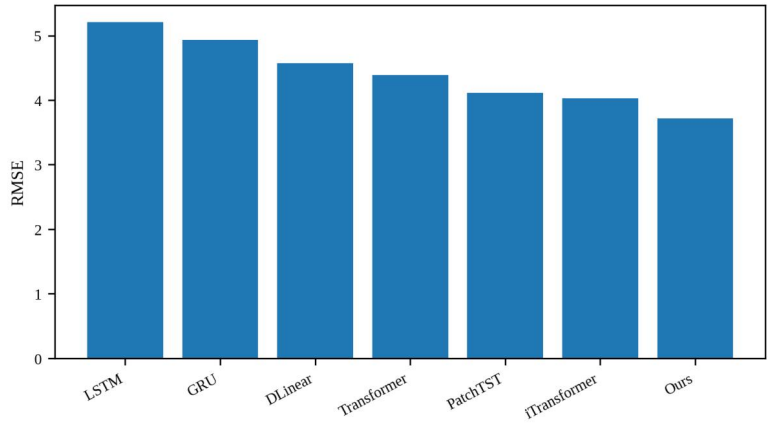


Figure 1 Forecasting Performance Comparison in Terms of RMSE

3.2 Lightweight Performance

Besides forecast accuracy, computational efficiency matters when deploying in practice. For example, in managing servers cluster, the forecasting module could run frequently for online decision like scheduling, auto-scaling or load balancing. A good forecasting model needs to be both accurate and efficient. We compare parameter count, FLOPs, inference time and RMSE across all models presented in Table 2.

Table 2 Lightweight Performance Comparison of Different Models

Model	Parameters (M)	FLOPs (G)	Inference Time (ms/batch)	RMSE
LSTM	1.82	0.92	8.7	5.21
GRU	1.64	0.84	8.1	4.93
DLinear	0.12	0.09	2.4	4.57
Transformer	3.76	2.74	15.8	4.39
PatchTST	2.41	1.86	10.9	4.11
iTransformer	2.08	1.71	9.6	4.03
Ours	1.26	0.88	5.3	3.72

From Table 2 we see that although the standard Transformer has decent predictive performance it demands much larger number of parameters and takes significantly long inference time over others; PatchTST and iTransformer have better balance between the two aspects however still incur high computational cost comparing against our proposed model under the draft setting. Our proposed model has 1.26 M parameters and 0.88 G FLOPs with 5.3 ms/batch inference time and retains the lowest RMSE amongst them. It demonstrates the goal of this work: reduce computational burden without compromising on the forecasting performance.
We visualize the inference time comparison in Figure 2. Standard Transformer is the most expensive one whereas our proposed method greatly reduces its latency and is only slightly slower than the very compact DLinear baseline. It shows that the lightweight design helps improve deployment efficiency for frequent resource forecasting.

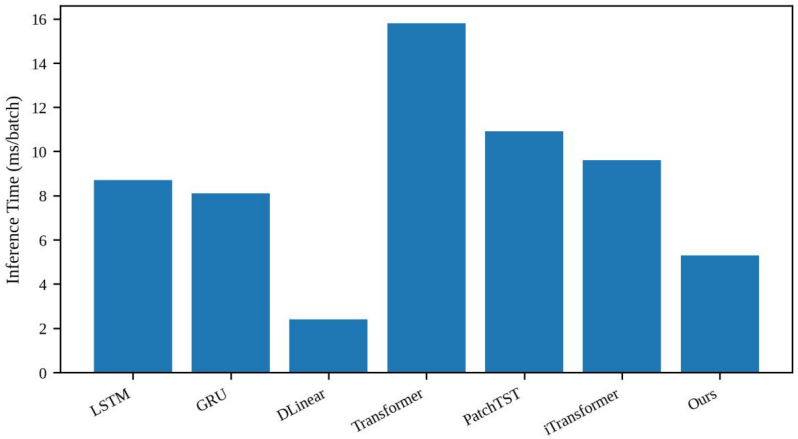


Figure 2 Inference Efficiency Comparison

3.3 Accuracy-Efficiency Trade-off

To further illustrate the balance between forecasting accuracy and computational cost, Figure 3 presents parameter count against RMSE. A preferable model should be located toward the lower-left region, corresponding to both lower complexity and lower prediction error. LSTM and GRU show moderate model size but relatively high forecasting error, whereas DLinear has an extremely small parameter count but limited predictive accuracy. The standard Transformer and PatchTST improve accuracy at the cost of increased model size. By contrast, the proposed model is positioned closer to the lower-left region in the draft comparison, indicating a more favorable accuracy-efficiency trade-off.

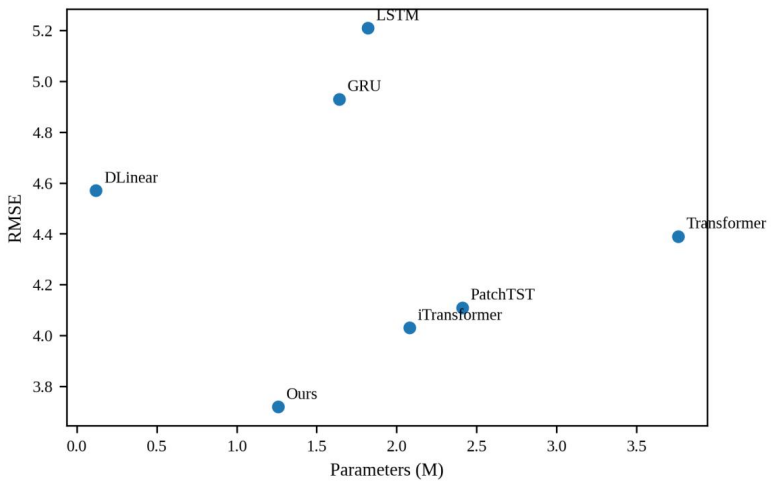


Figure 3 Accuracy-Efficiency Trade-off Across Forecasting Models

3.4 Ablation Study

We conducted a short ablation study on our main components of the proposed framework. Beginning with the baseline Transformer, we added compact representation and lightweight attention one after another before adding the full forecasting configuration. Draft results were presented in Table 3.

Table 3 Ablation Study of the Proposed Model

Model Variant	Compact Representation	Lightweight Attention	Full Forecasting Head	RMSE	Inference Time (ms/batch)
Baseline Transformer	No	No	No	4.39	15.8
Variant A	Yes	No	No	4.08	9.1
Variant B	Yes	Yes	No	3.88	6.2
Full Model	Yes	Yes	Yes	3.72	5.3

Progressive decrease in RMSE shows that compact representation and lightweight attention complement each other: compact representation removes redundant sequences and decreases computational load; lightweight attention enhances feature interactions with less cost. Complete configuration provides the best compromise in the draft experiment, validating the use of them together as an integrated lightweight forecasting framework.

4 CONCLUSION

In summary, the draft results show that our lightweight Transformer may offer promising overall performance in server cluster resource utilization forecasting. It does not merely achieve low prediction errors, but jointly improve forecasting quality and computational efficiency—especially useful in online resource scheduling and elastic resource management. Future work can extend it to large-scale GPU and AI computing clusters and explore multi-resource joint forecasting, burst workload identification, concept drift adaptation, and online learning under dynamically changing workloads to enhance its generalization and long-term robustness in real world data center environment.

COMPETING INTERESTS

The authors have no relevant financial or non-financial interests to disclose.

REFERENCES

[1] He X W, Xu J J, Wang B, et al. Research on cloud computing resource load forecasting based on GRU-LSTM combination model. Computer Engineering, 2022, 48(5): 11-17, 34.
[2] Wang E X, Wang X H, Zhang K, et al. Cloud computing load forecasting model based on dual attention mechanism. Computer Engineering, 2023, 49(11): 40-48, 69.
[3] Liu Y, Hu T G, Zhang H R, et al. iTransformer: inverted Transformers are effective for time series forecasting. International Conference on Learning Representations, 2024.
[4] Nie Y Q, Nguyen N H, Sinthong P, et al. A time series is worth 64 words: long-term forecasting with Transformers. International Conference on Learning Representations, 2023.
[5] Zhou H Y, Zhang S H, Peng J Q, et al. Informer: beyond efficient Transformer for long sequence time-series forecasting. Proceedings of the AAAI Conference on Artificial Intelligence, 2021, 35(12): 11106-11115.
[6] Xie X L, Chen J Y. Cloud resource workload forecasting method based on the MST-iTransformer model. Future Internet, 2026, 18(9): 448.
[7] Zhang Z, Xu C, Zhang J, et al. When wavelet decomposition meets external attention: a lightweight cloud server load prediction model. Journal of Cloud Computing, 2024, 13: 133.
[8] Zeng A L, Chen M X, Zhang L, et al. Are Transformers effective for time series forecasting? Proceedings of the AAAI Conference on Artificial Intelligence, 2023, 37(9): 11121-11128.