Article

# Temporal-Spatial Deep Learning for Memory Usage Forecasting in Cloud Servers

Kai Aidi and Danyi Gao *

Posted Date: 15 May 2025

doi: 10.20944/preprints202505.1223.v1

Keywords: cloud server; memory usage prediction; convolutional neural network; long short-term memory network

*Article*

# Temporal-Spatial Deep Learning for Memory Usage Forecasting in Cloud Servers

**Aidi Kai and Danyi Gao ***

[1] The University of Texas at Austin, Austin, TX, USA

[2] Columbia University, New York, NY, USA

* Correspondence: dg3224@columbia.edu

**Abstract:** To address the challenge of highly volatile and difficult-to-predict memory usage in cloud servers, this paper proposes a memory usage prediction model that integrates Convolutional Neural Networks (CNN) and Long Short-Term Memory networks (LSTM). The approach extracts local spatial correlations from input feature sequences through the CNN module and captures temporal dependencies using the LSTM structure. This enables high-precision prediction of memory usage trends over time. To validate the model's effectiveness, a prediction dataset was constructed using real-world cloud server monitoring data, covering ten key resource indicators. Comparative experiments were conducted with several mainstream deep learning models. The results show that the proposed CNN+LSTM model outperforms traditional models in terms of MSE, MAE, and $R^2$ metrics, demonstrating stronger fitting capability and greater stability. Loss convergence analysis and prediction curve comparisons further confirm that the model effectively captures the actual fluctuation patterns of resource usage. It performs particularly well on complex nonlinear sequences, exhibiting both strong predictive performance and practical engineering value.

**Keywords**: cloud server; memory usage prediction; convolutional neural network; long short-term memory network

## 1. Introduction

The widespread adoption of cloud computing has driven the virtualization and elastic expansion of computing resources, making cloud servers a critical component in supporting large-scale service deployments. As a core part of cloud platforms, the efficient scheduling and accurate prediction of server resources directly affect system performance, cost control, and service quality. Among various resources, memory usage plays a key role in task scheduling, load balancing, and resource reclamation, and has thus become a central focus in resource prediction research. However, due to multi-tenant concurrency and bursty service requests, memory usage in cloud servers often exhibits highly nonlinear and dynamic characteristics [1]. Traditional linear modeling methods struggle to capture such complex temporal evolution patterns, leading to significant prediction errors and impairing the accuracy and responsiveness of scheduling strategies. Therefore, it is of practical importance to develop prediction algorithms that can handle multidimensional inputs and accurately learn temporal dependencies.

In actual cloud platform scheduling systems, backend services often rely on resource usage status to execute service orchestration, container migration, or elastic scaling operations. These operations are highly sensitive to memory usage forecasts. For instance, when a node is predicted to reach a memory bottleneck, the system must promptly trigger migration strategies to alleviate resource pressure and prevent service interruptions or performance degradation. Thus, building a resource prediction model that can be directly integrated into backend systems not only enhances the intelligence of scheduling algorithms but also strengthens the system's adaptability under heavy load or abnormal fluctuations. Compared with static threshold configurations and experience-based tuning strategies, data-driven deep learning methods can autonomously learn inherent patterns from historical sequences, providing more forward-looking decision support for backend services.

With the advancement of deep learning in time-series modeling, increasing efforts have been made to combine Convolutional Neural Networks (CNN) and Long Short-Term Memory networks (LSTM) for complex sequence modeling tasks [2]. CNN excels in extracting local spatial features, capturing correlations among resource usage data across different dimensions. LSTM, on the other hand, is effective in modeling long-term dependencies and the sequential evolution of data. The combination of CNN and LSTM has demonstrated strong performance across various prediction tasks and is particularly suitable for handling time-series data with nonlinear fluctuations and multidimensional coupling, such as server resource usage [3]. By constructing a hybrid CNN+LSTM architecture, high-level abstract features can be extracted at the front end, while the backend models temporal dynamics to enable accurate prediction of memory usage.

From a system implementation perspective, the CNN+LSTM architecture offers good deployability and scalability, allowing seamless integration with existing monitoring systems, scheduling engines, and container orchestration platforms. Specifically, the prediction module can operate as an independent microservice within the backend, periodically retrieving historical server monitoring data to perform rolling predictions and dynamic evaluations of resource usage. The model also supports a variety of input indicators, such as CPU usage, disk I/O rates, network traffic, and average load, enhancing prediction accuracy and generalization. For backend engineering systems, such a model with real-time prediction capabilities can significantly reduce the frequency of manual intervention, improve resource utilization efficiency, and promote the development of cloud platforms toward greater adaptability and lower energy consumption. In summary, accurate prediction of cloud server memory usage is not only a core technical challenge in resource management but also a foundational capability for enabling intelligent backend scheduling and elastic resource control. As cloud infrastructure becomes the backbone of large-scale applications such as recommendation systems [4]and large language models (LLMs) [5], efficient memory forecasting becomes critical to sustaining real-time responsiveness, optimizing compute allocation [6,7], and maintaining service reliability under dynamic workloads. The CNN+LSTM-based prediction method proposed in this paper aims to establish a high-accuracy, highly generalizable prediction framework to address the real-world challenges of resource fluctuation and dynamic service demand in cloud environments. This approach is of great significance for enhancing the availability, stability, and resource efficiency of cloud platforms while also providing theoretical foundations and practical solutions for building intelligent cloud infrastructures.

## 2. Method

This study constructs a hybrid deep model that integrates convolutional neural networks (CNN) and long short-term memory networks (LSTM) to achieve dynamic prediction of cloud server memory usage. The overall structure consists of two parts: one is a CNN-based feature extraction module, which is mainly used to learn the local dependencies between various resource dimensions in the input sequence; the other is a LSTM-based time series modeling module, which aims to capture the long-term dependencies and evolution trends of feature sequences. The model architecture is shown in Figure 1.

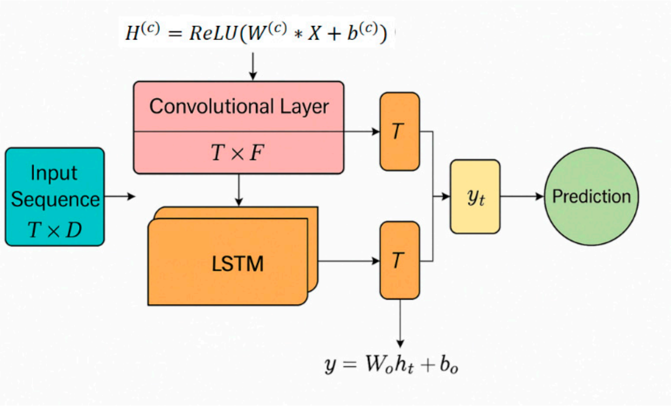

**Figure 1.** Overall architecture diagram.

The model begins by processing a multi-dimensional sequence of cloud resource metrics through a one-dimensional convolutional layer. This layer is designed to extract local spatial features and enhance sensitivity to inter-resource correlations, leveraging the strength of convolutional structures in handling high-dimensional and sparse data representations, as established in recent deep feature mining frameworks [8]. The resulting feature sequence is then passed into an LSTM module, which captures the dynamic patterns and long-term dependencies present in temporal resource behavior. This integration reflects practices in recent policy-driven scheduling systems, where temporal modeling contributes significantly to adaptive and stable decision-making in volatile environments [9]. The final stage of the model involves a fully connected layer that maps the learned feature representations to precise memory usage predictions. This end-to-end architecture incorporates principles from modern reinforcement learning-based scheduling designs, particularly those that demonstrate the effectiveness of hybrid deep networks in modeling complex resource usage trends under fluctuating cloud workloads [10]

The model input is a multi-dimensional resource monitoring sequence of length T, and the output is the predicted value of memory usage at the future moment. By jointly optimizing the two-part network structure, the model can simultaneously consider local features and temporal dynamic characteristics, effectively improving the prediction performance. Let the original input be $X \in R^{T \times D}$ , where T represents the time step length and D is the number of resource features contained in each time step. First, input X into the one-dimensional convolutional layer and perform the following feature extraction operations:

$$H^{(c)} = RELU(W^{(c)} * X + b^{(c)})$$

$*$ represents the convolution operation, $W^{(c)}$ and $b^{(c)}$ are the convolution kernel and bias respectively, and ReLU is the activation function. The convolution output $H^{(c)} \in R^{T \times F}$ , where F is the number of convolution kernels, represents the extracted high-dimensional features.

Then the convolution feature sequence is input into the LSTM network for time series modeling. The core state of the LSTM unit is updated as follows:

$$f_t = \sigma(W_f h_{t-1} + U_f x_t + b_f)$$

$$c_t = f_t \otimes c_{t-1} + i_t \otimes \tanh(W_c h_{t-1} + U_c x_t + b_c)$$

$$h_t = o_t \otimes \tanh(c_t)$$

$f_t, i_t, o_t$ is the forget gate, input gate and output gate, B is the unit memory state, and $c_t$ is the hidden state at the current moment. $h_t$ at the final moment is used as the representation feature of the sequence and input into the fully connected layer to predict the memory usage.

The output layer adopts the linear mapping form:

$$y' = W_o h_T + b_o$$

where $y'$ is the predicted value, $W_o$ and $b_o$ are weights and bias terms. During the training process, the mean square error (MSE) is used as the loss function:

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

The model minimizes the above losses through the Adam optimizer and continuously adjusts the parameters of each layer to converge to the optimal state. The overall process builds a cloud server memory prediction framework with strong generalization capabilities based on both spatial characteristics and time dependencies.

## 3. Experiment

### 3.1. Datasets

The dataset used in this study consists of 5,000 records collected from actual cloud server runtime environments. It includes ten key resource usage indicators such as CPU utilization, memory usage, disk I/O rate, network transfer rate, number of active processes, and system load. All data were obtained through continuous monitoring under high-concurrency service conditions, ensuring both temporal dynamics and representativeness. This allows the dataset to reflect realistic resource usage patterns in cloud computing scenarios.

To enhance modeling performance and generalization ability, the data underwent necessary cleaning and preprocessing steps after collection. These included removing duplicate samples, filling missing values, and performing feature normalization. A unified data format and time window structure were established, making the dataset directly applicable for time-series deep learning models. The dataset satisfies the input requirements for predictive tasks. Moreover, it exhibits significant correlation and non-stationarity across both temporal and resource dimensions, which supports the learning of complex temporal dependencies.

The dataset demonstrates typical fluctuations in cloud server resource usage, such as periodic load variations, burst resource consumption, and multidimensional metric coupling effects. It is well-suited as a foundation for training and evaluating memory usage prediction models. Models built on this dataset can be used not only for performance validation but also for engineering deployment. They offer practical predictive insights to support real-world system scheduling.

### 3.2. Experimental Results

This paper first gives the comparative experimental results of different algorithms, and the experimental results are shown in Table 1.

**Table 1.** Comparative experimental results.

| Method | MSE | MAE | $R^2$ |
|---|---|---|---|
| MLP [11] | 0.0016 | 0.2513 | 0.51 |
| GRU [12] | 0.0009 | 0.1746 | 0.66 |
| LSTM [13] | 0.0006 | 0.1295 | 0.73 |
| CNN [14] | 0.0004 | 0.1027 | 0.76 |
| Ours(CNN+LSTM) | 0.0001 | 0.0513 | 0.82 |

The experimental results show that the traditional Multilayer Perceptron (MLP) performs relatively poorly in the task of memory usage prediction. Its MSE and MAE values are significantly higher than those of other methods, indicating its limited ability to capture complex temporal patterns and feature interactions in resource usage data. Although MLP possesses certain nonlinear modeling capabilities, its architecture restricts effective representation of historical state information when handling multidimensional time-series data, leading to large prediction errors.

In contrast, recurrent neural network models such as GRU and LSTM exhibit clear advantages in modeling temporal dependencies. Both models achieve better performance in error control and fitting accuracy. Notably, LSTM demonstrates strong stability in long-term dependency modeling, achieving an $R^2$ value of 0.73. This indicates its strong generalization ability in capturing memory usage fluctuation trends. Although CNN does not explicitly model time dependencies, it improves prediction accuracy through local perception and feature convolution, showing effectiveness in extracting high-dimensional resource features.

After integrating CNN and LSTM structures, the proposed CNN+LSTM model achieves the best performance across all evaluation metrics. It records the lowest MSE and MAE values, and reaches an $R^2$ value of 0.82. These results demonstrate that extracting local feature correlations via convolutional modules, combined with LSTM's ability to model temporal dynamics, significantly enhances the capacity to predict cloud server memory usage. The model's superior accuracy and stability provide reliable, data-driven support for real system deployment and confirm its practical value in cloud resource prediction scenarios.

Secondly, this paper provides a graph illustrating the decline of the loss function during the training process, as shown in Figure 2. The figure clearly shows the loss reduction trend over successive training epochs, offering an intuitive understanding of the model's convergence behavior. It serves as evidence of the model's learning efficiency and training stability throughout the optimization process.

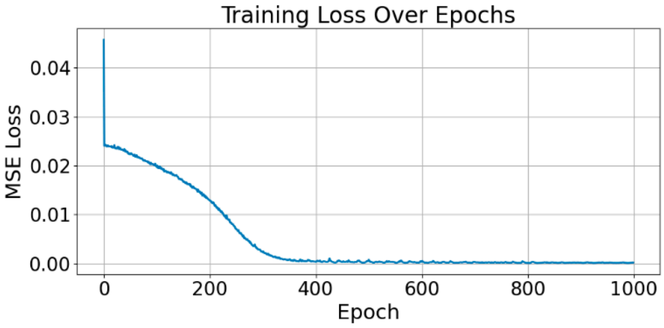

**Figure 2.** Loss function drop graph.

The training loss curve reveals rapid convergence in the initial phase, with the MSE dropping sharply within the first 100 epochs, indicating effective early learning of key patterns in cloud server memory usage. After approximately 400 epochs, the model enters a stable phase with the loss approaching zero, reflecting high fitting accuracy and consistent convergence without fluctuations. This smooth trajectory underscores the model’s architectural soundness and robust parameter updates, enabling it to learn complex temporal features without overfitting or instability. Throughout the 1000 epochs, the model maintains low loss values, affirming its strong generalization and predictive reliability—critical for dynamic and elastic resource management in cloud environments. As shown in Figure 3, predicted values closely align with actual memory usage, further validating the model’s forecasting accuracy.

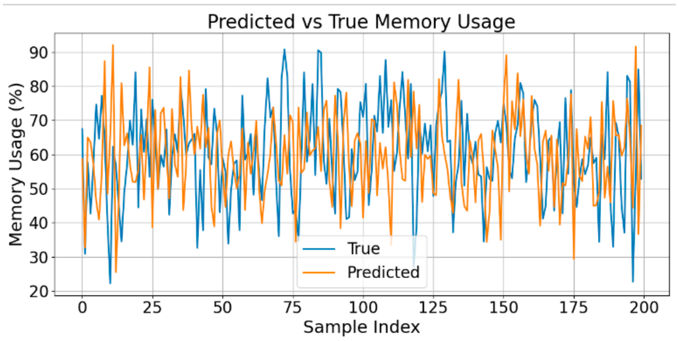

**Figure 3.** Comparison between actual and predicted values.

As shown in the figure, the predicted curve aligns closely with the actual memory usage curve in terms of overall trend. The model effectively captures the main fluctuation direction and variation patterns of memory usage. Notably, even in regions with sharp changes, the predicted values follow the actual values closely, demonstrating the model's ability to adapt to abnormal variations in dynamic environments.

Although some prediction deviations exist, the errors are mainly concentrated around localized peaks and valleys where changes occur rapidly. These discrepancies may be caused by instantaneous disturbances in the input features or sparse temporal information. However, for most data points, the predicted values remain close to the actual values. This indicates that the CNN+LSTM model maintains good global fitting stability and response capability. Overall, the results confirm the model's practicality in cloud server resource management. It can provide useful forecasts of memory usage trends for backend systems. In real-world deployment, such predictive capability supports elastic resource scaling and fault warning, thereby improving the overall efficiency and reliability of the system.

## 4. Conclusions

This research paper explores a crucial aspect of cloud server resource management: predicting memory usage. It introduces a sophisticated deep learning model that combines convolutional neural networks (CNNs) and long short-term memory (LSTM) networks to tackle this challenging task. The model extracts local feature correlations through convolutional layers and captures temporal dependencies in sequences using long short-term memory structures. This enables high-precision modeling of complex resource dynamics. Experimental results confirm that the proposed method outperforms traditional models across multiple performance metrics, demonstrating superior predictive accuracy and strong generalization ability.

The results show that the model converges quickly during training and produces stable, accurate forecasts of future memory usage trends. It performs particularly well when handling highly nonlinear and volatile sequence data. These characteristics make it valuable for deployment in high-concurrency, multi-tenant environments, effectively supporting intelligent scheduling and resource optimization strategies.

In addition, the proposed framework exhibits strong scalability and integrability. It can be seamlessly embedded into existing backend management systems and used with real-time monitoring data for rolling prediction. By modeling multiple resource indicators in a coordinated manner, this method offers a viable path toward building adaptive, low-latency cloud infrastructure. It also provides theoretical support for the development of intelligent operation and maintenance systems. Future research may extend the applicability of the model to multi-task and multi-node scenarios by integrating federated learning, adaptive optimization, or attention mechanisms. This would enhance the model's generalization and interpretability. Moreover, incorporating heterogeneous resource features and service context information may lead to the development of more context-aware predictive systems, further advancing cloud platforms toward intelligent autonomy and efficient resource scheduling.

## References

1. Md N. Newaz and Md A. Mollah, "Memory usage prediction of HPC workloads using feature engineering and machine learning", Proceedings of the International Conference on High Performance Computing in Asia-Pacific Region, 2023.
2. F. C. Minuzzi and L. Farina, "A deep learning approach to predict significant wave height using long short-term memory", Ocean Modelling, vol. 181, Art. no. 102151, 2023.
3. R. Huang, et al., "Well performance prediction based on Long Short-Term Memory (LSTM) neural network", Journal of Petroleum Science and Engineering, vol. 208, Art. no. 109686, 2022.
4. A. Liang, "Enhancing Recommendation Systems with Multi-Modal Transformers in Cross-Domain Scenarios", Journal of Computer Technology and Software, vol. 3, no. 7, 2024.
5. R. Wang, "Joint Semantic Detection and Dissemination Control of Phishing Attacks on Social Media via LLama-Based Modeling", 2025.
6. B. Wang, "Topology-Aware Decision Making in Distributed Scheduling via Multi-Agent Reinforcement Learning", Transactions on Computational and Scientific Methods, vol. 5, no. 4, 2025.
7. L. Zhu, F. Guo, G. Cai and Y. Ma, "Structured Preference Modeling for Reinforcement Learning-Based Fine-Tuning of Large Models", Journal of Computer Technology and Software, vol. 4, no. 4, 2025.
8. W. Cui and A. Liang, "Diffusion-Transformer Framework for Deep Mining of High-Dimensional Sparse Data", Journal of Computer Technology and Software, vol. 4, no. 4, 2025.
9. Y. Ren, M. Wei, H. Xin, T. Yang and Y. Qi, "Distributed Network Traffic Scheduling via Trust-Constrained Policy Learning Mechanisms", Transactions on Computational and Scientific Methods, vol. 5, no. 4, 2025.
10. Y. Wang, T. Tang, Z. Fang, Y. Deng and Y. Duan, "Intelligent Task Scheduling for Microservices via A3C-Based Reinforcement Learning", arXiv preprint, arXiv:2505.00299, 2025.
11. M. B. Taha, Y. Sanjalawe, A. Al-Daraiseh, S. Fraihat and S. R. Al-E’mari, "Proactive auto-scaling for service function chains in cloud computing based on deep learning", IEEE Access, vol. 12, pp. 38575–38593, 2024.
12. X. Li, et al., "Resource usage prediction based on BiLSTM-GRU combination model", Proceedings of the 2022 IEEE International Conference on Joint Cloud Computing (JCC), 2022.
13. J. Q. Wang, Y. Du and J. Wang, "LSTM based long-term energy consumption prediction with periodicity", Energy, vol. 197, Art. no. 117197, 2020.
14. E. Patel and D. S. Kushwaha, "A hybrid CNN-LSTM model for predicting server load in cloud computing", The Journal of Supercomputing, vol. 78, no. 8, pp. 1–30, 2022.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.