Transactions on Computational and Scientific Methods | Vo. 4, No. 5, 2024
ISSN: 2998-8780
https://pspress.org/index.php/tcsm
Pinnacle Science Press

Dynamic Resource Load Forecasting in Cloud Computing: A Representation Learning Perspective

Tianjian Xia
Syracuse University, Syracuse, USA
tj.hsia@gmail.com

Abstract: In cloud computing environments, resource loads exhibit strong temporal volatility, non-stationarity, and multi-source coupling. Accurate prediction of their evolution is a fundamental requirement for efficient resource management and service quality assurance. To address the limitations of traditional methods in capturing complex temporal dependencies and load generation mechanisms, this paper proposes a dynamic cloud resource load prediction approach based on deep time series representation learning. The method models multivariate historical resource monitoring data as unified time series inputs. Through temporal encoding and representation aggregation, it automatically learns key patterns in load variation. This enables a robust characterization of load evolution in latent space. Under a unified data setting, comparative analysis with several representative time series prediction models shows clear advantages in error control and overall stability. The results indicate that a representation-centered modeling paradigm effectively mitigates the impact of noise and distribution fluctuations. It is therefore more suitable for complex and dynamic load behaviors in real cloud environments. This study provides a systematic solution for cloud resource load prediction and offers valuable reference for intelligent resource allocation, capacity planning, and cloud platform optimization.

Keywords: Cloud resource load forecasting; time series modeling; deep representation learning; resource management optimization

1. Introduction

Under the deep integration of cloud computing and artificial intelligence, modern information infrastructures are evolving into highly dynamic systems with large-scale and complex structures. Cloud platforms provide on-demand services through virtualization and resource pooling. Their operational states are jointly influenced by user request patterns, application deployment strategies, and underlying resource scheduling policies. These dynamic characteristics cause cloud resource loads to exhibit strong temporal fluctuations and non-stationarity. Variations at different time scales are often superimposed, forming complex evolution processes. In this context, accurately characterizing the temporal evolution of cloud resource loads and reliably predicting future trends has become a fundamental issue for ensuring stable operation and efficient service delivery [1].

With the widespread adoption of cloud native architectures, microservice paradigms, and multi-tenant environments, the mechanisms that generate cloud resource loads have become increasingly complex. Load variations of a single service are no longer determined solely by its own request intensity. They are also affected by upstream and downstream service interactions, resource contention, and scheduling behaviors. This multi-source and multi-factor coupling leads to time series with high dimensionality, strong correlations,

and pronounced nonlinearity. Traditional methods based on statistical assumptions or simple regression relationships struggle to capture such structures. Moreover, cloud platforms inevitably experience bursty traffic, policy adjustments, and environmental disturbances. These factors further increase uncertainty and transform load prediction into a problem of understanding the overall dynamics of a complex system [2].

In cloud computing environments, dynamic resource load prediction has not only theoretical value but also direct practical importance. When load changes cannot be anticipated in advance, resource allocation often lags behind demand. This can cause performance degradation and service quality loss. In severe cases, cascading risks may emerge. By contrast, effective modeling of historical operational data enables early awareness of load evolution. This provides foresight for resource management and scheduling. Systems can then shift from reactive adjustment to proactive planning. Such a transition is critical for enhancing elasticity and operational resilience. It is also a key prerequisite for intelligent cloud resource management [3].

Recent advances in deep learning have demonstrated strong capabilities in modeling complex dependencies in time series data. This progress offers new opportunities for cloud resource load prediction. Instead of directly forecasting raw numerical values, deep time series representation learning focuses on extracting multi-scale temporal features and latent evolution patterns. It can build more robust representations under noisy and highly nonlinear conditions. This representation-centered paradigm helps reveal structural regularities behind load sequences [4], [5]. It therefore improves the understanding of future trends. Introducing deep time series representation learning into cloud load prediction can enhance stability. It also supports unified modeling across different resource types and diverse operational scenarios.

For these reasons, systematic exploration of time series modeling and prediction for dynamic cloud resource loads is of significant research interest. On the one hand, it deepens the understanding of cloud system operation mechanisms and load evolution patterns. This provides theoretical support for resource management in complex environments. On the other hand, constructing temporal representation models that adapt to dynamic changes and multi-source coupling lays a foundation for intelligent scheduling, capacity planning, and risk warning. As cloud computing applications continue to evolve, more forward-looking and systematic load prediction studies will play a long-term role in advancing intelligent and highly reliable cloud platforms [6].

2. Research Motivation

As cloud platforms continue to expand in scale and evolve in service forms, the volatility and complexity of resource loads have increased markedly. In real operations, cloud resource loads often exhibit a mixture of aperiodic fluctuations, sudden peaks, and long-term trends. Implicit correlations also exist across different resource dimensions. These complex evolution patterns make it difficult to rely on short-term history or a single temporal scale to form reliable judgments about future load changes [7]. As a result, the foresight of resource management strategies is constrained. Identifying temporal features that play a decisive role in load evolution from large-scale operational data has therefore become a central motivation in current cloud load prediction research.

From another perspective, many existing load prediction approaches lack holistic modeling capabilities when facing dynamic changes in cloud environments. During operation, cloud platforms frequently experience adjustments in service structures, changes in scheduling policies, and shifts in external access patterns. These factors lead to continuous distribution drift in load time series [8]. If prediction models depend excessively on fixed patterns or local statistical characteristics, their performance tends to degrade under changing conditions. This weakens the practical value of predictions for real resource planning. Consequently, developing prediction methods that can learn stable temporal representations and adapt to dynamic environments is a key driving force for advancing load prediction research [9].

In response to these challenges, the core motivation of this study is to explore a load modeling approach centered on deep time series representation learning. The aim is to capture common structural patterns of

cloud resource loads across different time scales and operating conditions. This enhances the model's ability to understand complex evolution processes. Compared with directly fitting numerical variations, representation-level modeling helps abstract the intrinsic patterns underlying load changes. It therefore improves robustness in long-term operation and complex scenarios. This motivation not only addresses practical needs in dynamic cloud load prediction but also provides methodological support for building more intelligent and adaptive cloud resource management systems.

Recent work on representation-based forecasting and adaptive recognition provides a broader methodological foundation for cloud load prediction. TrendFormer introduces two-stage trend fusion for financial time series forecasting, which supports the separation of persistent trends from short-lived fluctuations when modeling resource-load evolution [10]. Prior-enhanced representation learning under weakly labeled and few-shot conditions further shows how stable priors can improve recognition when labeled abnormal states are limited or unevenly distributed [11]. Unified multi-source feature learning and structural aggregation also indicates that heterogeneous signals can be fused into a common representation before downstream risk or load-state judgment [12]. These studies reinforce the need to model cloud resource loads as multi-source, non-stationary sequences rather than isolated metric curves.

Transformer-based structure-aware segmentation with multi-scale fusion and boundary-guided prediction offers a transferable view of how multi-scale representations and decision boundaries can be jointly strengthened in complex prediction tasks [13]. Retrieval-augmented long-text reasoning with semantic conflict detection and cross-paragraph evidence integration further suggests that distributed operational evidence can be organized to reduce contradictory interpretations across logs, monitoring records, and scheduling descriptions [14]. For cloud resource forecasting, these ideas support a representation-learning framework that integrates trend decomposition, multi-source feature aggregation, boundary-sensitive pattern separation, and evidence-consistent temporal modeling.

3. Model Design

To characterize the temporal evolution of dynamic cloud resource load, the multi-dimensional resource monitoring data collected during cloud platform operation is first represented as a unified time series format. Let $x_t \in R^d$ be the resource state vector observed at time step t, where d represents the number of resource dimensions, such as compute, storage, and network. Given a historical observation window of length T, the input sequence can be represented as:

$$X_{1:T} = \{x_1, x_2, \ldots, x_T\}$$

This representation can uniformly describe the joint changes of different resource dimensions over time, providing structured input for subsequent time-series representation learning. Based on this, by using sliding modeling of historical windows, the model can continuously perceive the dynamic changes in cloud resource load.

To capture the temporal dependencies inherent in the load sequence, a hidden state modeling mechanism based on recursive mapping is introduced to map the original input to the latent representation space. The updated form of the hidden state $h_t \in R^k$ at time step t is defined as:

$$h_t = \phi(W_h h_{t-1} + W_x x_t + b_h)$$

Here, W_h and W_x are learnable weight matrices, b_h is a bias term, and $\phi(\bullet)$ represents a nonlinear mapping function. This recursive structure can accumulate historical information over time, allowing the hidden state to remain sensitive to recent changes while gradually encoding long-term load evolution characteristics, thus forming a continuous representation of the cloud resource operating status. This paper presents the overall model architecture, which is shown in Figure 1.

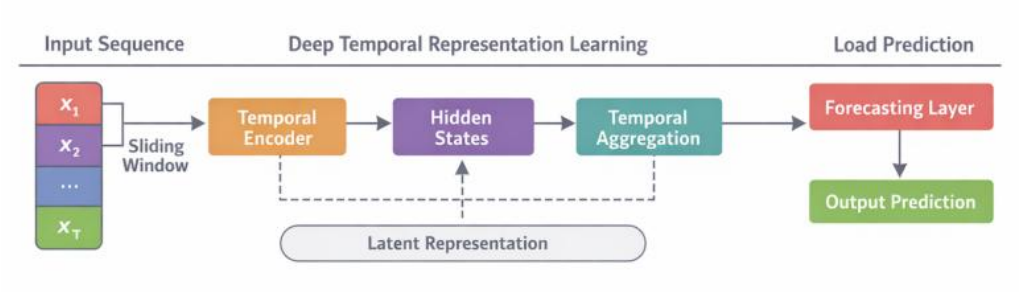

Figure 1. Overall model architecture

Considering that cloud resource load exhibits both short-term fluctuations and long-term trends, the final time-series representation is defined as the aggregation result of the hidden state sequence. Specifically, the hidden states within the time window are weighted and aggregated to obtain the global representation vector z:

$$z = \frac{1}{T}\sum_{t=1}^{T} h_t$$

This representation can reduce the impact of instantaneous noise on the overall modeling while maintaining the integrity of time series information, allowing the model to focus more on the overall structural features of load changes. Through this representation learning method, complex multivariate time series are mapped into discriminative low-dimensional representations, providing stable input for subsequent prediction tasks.

In the forecasting phase, future resource load is estimated based on the learned time-series representation. Let the forecast step size be τ, then the load forecast result $T + \tau$ at time $y_{T+\tau}$ is defined as:

$$y_{T+\tau} = W_o z + b_o$$

Where W_o and b_o are the output layer parameters. To guide the model in learning an effective temporal representation, the overall optimization objective is achieved by minimizing the deviation between the predicted value and the actual load, and its loss function is as follows:

$$L = \frac{1}{N}\sum_{i=1}^{N} \| \hat{y}^{(i)} - y^{(i)} \|_2^2$$

Where N represents the number of samples and $y^{(i)}$ represents the corresponding actual load value. Through the above modeling process, the method completes the temporal representation learning and predictive modeling of dynamic cloud resource load within a unified framework, providing a systematic solution for resource evolution modeling in complex cloud environments.

4. Implementation

4.1 Dataset

This study uses the Alibaba Cluster Trace as a public benchmark dataset for dynamic cloud resource load prediction. The dataset is released by Alibaba and is collected from monitoring and scheduling logs of real production clusters. It covers resource usage behaviors of large-scale online services and batch jobs. The data exhibits typical cloud platform characteristics. These include strong load volatility, the coexistence of periodic patterns and bursty events, and resource contention and coupling among tasks. These properties

make the dataset highly suitable for research on dynamic cloud load prediction and deep time series representation learning.

In terms of content, the Alibaba Cluster Trace provides multi-granularity time series records. Commonly used dimensions include CPU utilization and memory usage at the task or container level, resource requests and allocations, task state transitions, and scheduling and deployment events. In research settings, a single container or task can be treated as a prediction target. Its time-varying resource utilization forms a multivariate time series input. Resource usage can also be aggregated at the machine level to construct node-level load sequences. This setting better matches capacity planning and elastic scaling scenarios. Since the data comes from real operational environments, it contains irregular fluctuations, intermittent missing values, and state-switching segments. These characteristics offer challenging and realistic signals for time series representation learning.

To support load prediction modeling, raw logs and monitoring records can be aligned to a unified time resolution through fixed interval resampling. Missing segments can be handled using standard treatments such as forward filling or truncation of invalid intervals. Different resource dimensions are then normalized to reduce scale differences. Samples are constructed using a sliding window strategy. Past multivariate resource sequences are used as inputs to predict the target load at a future time step, such as a CPU-related load metric. To ensure strict time series forecasting, data splitting follows the temporal order. Early periods are used for training, intermediate periods for validation, and later periods for testing. This prevents information leakage and more accurately reflects generalization performance in cloud environments.

4.2 Specific implementation

In the implementation, key fields related to resource load are first extracted from the original monitoring and scheduling records. Data from different sources are then aligned to a fixed time granularity to form a continuous multivariate time series input. Each prediction object can be selected as a single container, task, or node, and a feature vector sequence composed of metrics such as CPU, memory, network, and disk is constructed for it. Missing values and outliers are handled using consistent regularization to ensure the usability and stability of the sequence. Subsequently, the features of each dimension are normalized or standardized to reduce the impact of differences in scale on model training. T sliding window is used to generate sample pairs, using historical sequences of length A as input to predict the load target for the next τ steps, thereby transforming the original runtime data into supervised prediction data suitable for deep time series learning.

The model employs a pipelined structure of "temporal encoding—representation aggregation—prediction mapping" for training and inference: the temporal encoding module extracts temporal dependencies and change patterns from historical windows to obtain time-step hidden representations; the representation aggregation module further integrates information within the window to form a more stable global temporal representation; the prediction layer maps the representation to future load outputs, enabling multi-step or single-step predictions of target resource indicators. During training, regression loss is the core optimization objective, supplemented by strategies such as batch training, gradient pruning, and early stopping to improve convergence stability; the inference phase uses the same sliding window approach to generate prediction results on the timeline, and the output can be directly used for subsequent resource planning and scheduling decision support, ensuring reproducibility and deployability in the data processing, modeling, and prediction output stages.

5. Experimental Results and Analysis

This paper first presents the experimental results compared with other models, as shown in Table 1.

Table 1: Comparative experimental results

Method	MSE	MAE	MAPE	RMSE
LSTM [15]	0.842	0.611	8.72	0.917
BILSTM [16]	0.801	0.589	8.41	0.895
Transformer [17]	0.764	0.552	8.02	0.874
TimeMixer [18]	0.721	0.533	7.88	0.849
Informer [19]	0.689	0.518	7.71	0.829
Ours	0.612	0.471	7.29	0.782

The comparative results show a clear and continuous improvement from traditional recurrent structures to long sequence modeling approaches. This trend indicates that cloud resource load time series strongly require the modeling of long-term dependencies and the extraction of global patterns. Models based on recurrent units can capture local continuous variations. However, under bursty fluctuations and cross-scale trend superposition that are common in cloud environments, they often fail to fully exploit distant temporal information. Attention-based models perform better in such scenarios. This highlights the critical role of global context modeling in dynamic load prediction.

Relative changes in error metrics demonstrate consistent advantages across different evaluation criteria. This suggests that the improvement is not limited to a small number of samples or specific intervals. It contributes to overall prediction stability and bias control. For cloud load prediction, such consistency is particularly important. Resource management typically considers both average error and the reliability of fluctuation ranges. Improvements in one metric accompanied by degradation in another can lead to unstable capacity planning or elastic scaling decisions. The current results show a synchronized reduction in overall error levels. This better aligns with the demand for robust predictions in cloud resource management.

Compared with multiple long sequence forecasting baselines, the method still achieves lower errors. This indicates a more sufficient capture of complex load evolution at the representation learning level. Cloud resource loads are driven by multiple factors. They include periodic and trend components, as well as non-stationary disturbances caused by scheduling policies and multi-tenant competition. Models that rely only on explicit sequence mapping or fixed pattern fitting are prone to bias under distribution shifts or sudden changes. The results indicate that the adopted deep temporal representation mechanism is more effective at integrating multi-scale information and suppressing noise. This enhances the ability to characterize complex dynamic processes.

Overall, the experimental comparisons support the core objective of this study. The goal is to achieve more robust and generalizable time series prediction for dynamic cloud resource loads in complex environments. Lower overall errors imply a more accurate characterization of future load trends. This helps establish a more reliable forward-looking link between resource supply and demand. For tasks such as capacity planning, elastic scaling, and risk warning in cloud platforms, more stable predictions can reduce cost fluctuations caused by delayed strategies or over-provisioning. They also mitigate performance degradation risks induced by prediction bias. These findings demonstrate the application value of the method for intelligent cloud resource management.

The input window length determines the range of historical context that the model can utilize during prediction, thus affecting its ability to characterize cloud resource load evolution patterns. To examine the method's dependence on and stability of the historical window setting, a sensitivity visualization based on

different input window length configurations is presented below. The experimental results are shown in Figure 2.

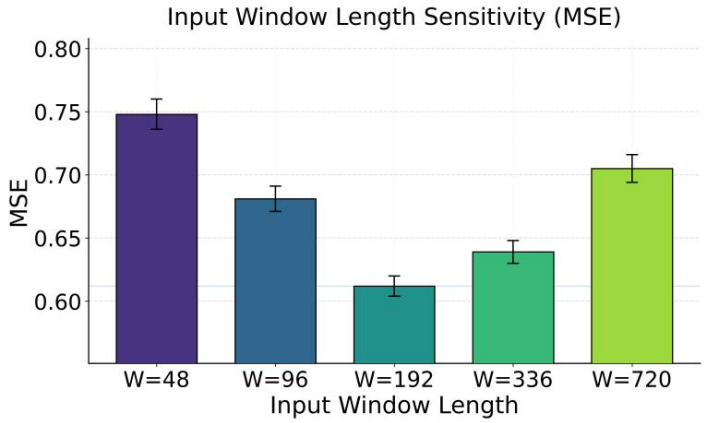

Figure 2. Sensitivity experiment of input window length to MSE

From an overall perspective, the effect of input window length on dynamic cloud load prediction errors shows a clear non-monotonic pattern. This indicates that the task does not become more accurate simply by using longer historical inputs. With short windows, the model mainly relies on local context to fit recent fluctuations. It is therefore sensitive to bursty spikes and short-term noise. This limits its ability to capture the overall shape of load evolution. As the window length increases, the model observes a more complete operational trajectory. It can better capture periodic behaviors and stage-level trends. Prediction bias is then effectively reduced.

Within a moderate window range, errors reach a relatively optimal level. This suggests a better balance between information sufficiency and temporal redundancy. Cloud platform loads are driven by multiple factors. They include stable daily rhythms and non-stationary disturbances caused by scheduling and multi-tenant competition. A moderate historical range can cover key change segments and trend turning points. At the same time, it avoids introducing excessive history that may cause distribution drift or irrelevant pattern interference. This allows deep temporal representation learning to focus more on core structural features related to load generation mechanisms.

When the window is further extended, errors increase again. This reflects the negative effects of excessive historical information. The statistical properties of cloud resource loads drift over time. Distant historical segments often correspond to different service stages or scheduling conditions. If they are directly encoded, the model may be forced to explain conflicting operating patterns. This weakens representations that are specific to the current state. In addition, long windows can amplify the impact of missing value imputation, monitoring jitter, and residual anomalies. Representation learning is then diluted by noise and redundancy. This ultimately leads to higher prediction errors.

The overall error bars are relatively short and show limited variation. This indicates a certain degree of output stability across different window settings. However, window selection still has a significant impact on performance upper bounds and robustness. For cloud resource management, this observation has direct implications. Window length reflects the degree of dependence on historical context. It therefore affects the reliability of decisions related to elastic scaling, capacity planning, and risk warning. These sensitivity results further emphasize that representation learning based modeling should align with the temporal scale characteristics of cloud environments. Selecting an appropriate historical range enhances the modeling of load evolution patterns. It also provides a more robust forward-looking basis for subsequent resource scheduling and governance.

The hidden dimension determines the capacity and expressive power of the temporal representation space, thus affecting the granularity with which the model characterizes the evolution patterns of cloud resource

load. To evaluate the impact of representation capacity settings on prediction stability and generalization, the following section visualizes the trend of model output error under different hidden dimension configurations. This analysis helps determine a more suitable representation size choice between information compression and pattern fidelity. The experimental results are shown in Figure 3.

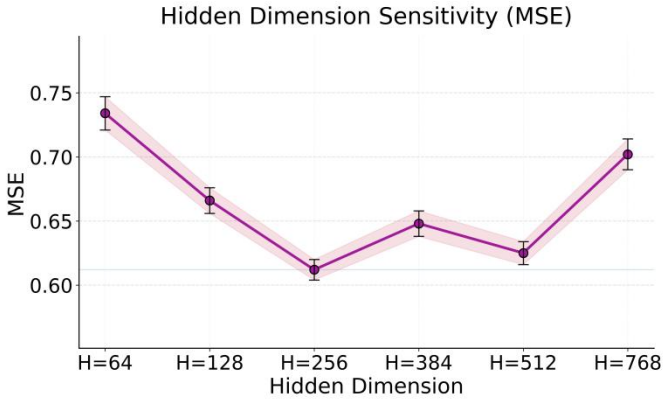

Figure 3. Sensitivity experiment of hidden dimensions to MSE

The hidden dimension has a clear impact on prediction errors for dynamic cloud resource loads. This highlights the critical role of representation capacity in time series modeling. With small hidden dimensions, the model has a limited ability to describe complex load evolution patterns. It cannot jointly express multi-scale fluctuations and latent trend structures. Prediction errors, therefore, remain at a relatively high level. This indicates that low-capacity representations are insufficient in cloud environments, where complex temporal information arises from multi-tenant competition, scheduling behaviors, and load coupling.

As the hidden dimension increases, errors decrease significantly. This shows that richer representation spaces help capture key variation patterns in cloud resource loads. At this stage, the model achieves a more reasonable balance between information compression and feature fidelity. Temporal representations are neither overly simplified nor dominated by noise. This observation aligns with the generation mechanisms of cloud loads. Load evolution is usually driven by multiple latent factors. Sufficient representation dimensions are required for effective modeling.

When the hidden dimension is further enlarged, errors begin to increase again. This suggests that excessive representation capacity does not guarantee continued improvement. An overly large hidden space may introduce redundant features. The model may then fit multiple weakly related or even conflicting patterns. This reduces its focus on core load structures. For cloud time series with strong non-stationarity, such overexpression can amplify historical noise and stage-specific differences. This can interfere with predictions of the current state.

The overall range of error bands remains relatively stable. This indicates a certain level of robustness across different hidden dimension settings. However, the upper bound of performance still depends on a reasonable choice of representation scale. These sensitivity results further emphasize that in deep time series representation learning for cloud load prediction, the hidden dimension is more than a capacity parameter. It directly affects the quality of abstraction of load evolution patterns. Selecting an appropriate representation scale enables a better balance between expressive power and generalization stability. This provides more reliable predictive support for capacity planning and elastic decision-making in cloud resource management.

Finally, the effect of batch size on the experimental results is presented, and the results are shown in Figure 4.

Batch size has a direct influence on the optimization dynamics of cloud load prediction, and this effect is reflected in the prediction errors under different batch settings. When the batch size is small, gradient estimates are dominated by sampling noise, and parameter updates become more sensitive to transient fluctuations in the training sequences. For dynamic cloud resource loads, where the series often contains

bursty segments and stage-wise distribution shifts, overly noisy updates can make the model chase short-term irregularities rather than stable temporal structures. As a result, the overall error level tends to remain relatively high, and the performance becomes less stable.

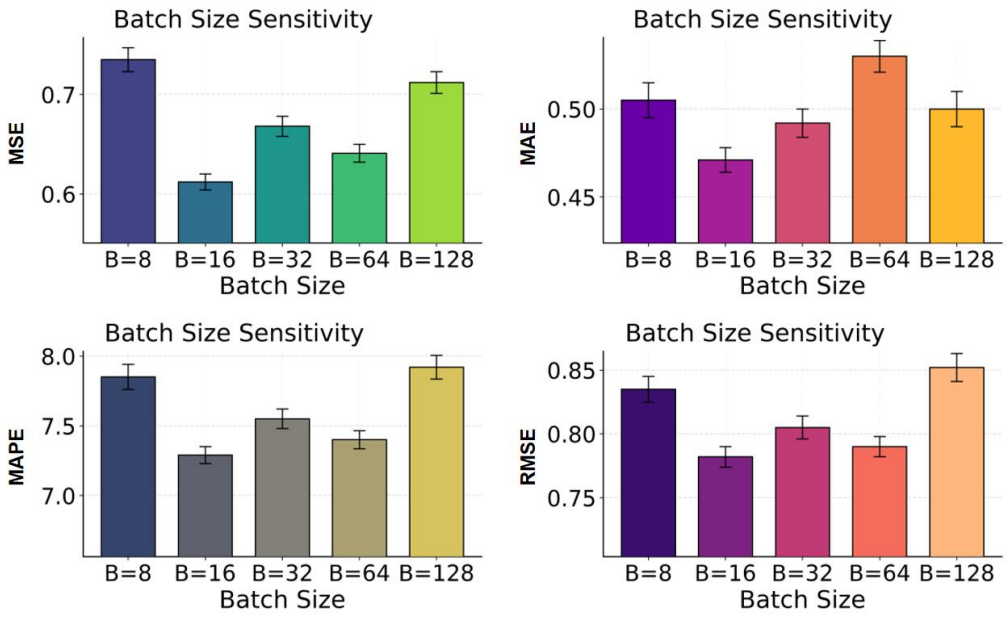

Figure 4. The effect of batch size on experimental results

As the batch size increases to a moderate range, errors typically decrease, and the results become more consistent. This indicates that a more reliable gradient signal helps the model learn representations that better capture the underlying evolution patterns of cloud loads. In this regime, the training process benefits from improved statistical efficiency: mini-batches contain more diverse temporal contexts, reducing the probability that a single bursty window or a local anomaly dominates the update direction. Consequently, the model reaches a better balance between fitting local fluctuations and preserving global trend information, which is essential for cloud environments affected by multi-tenant contention, scheduling perturbations, and load coupling.

However, further enlarging the batch size does not necessarily yield continuous improvement, and errors may rise again when the batch becomes too large. Large batches often reduce the stochasticity, which can be beneficial for escaping sharp minima and can lead to weaker generalization on non-stationary time series. In cloud workloads, the data distribution may vary across time periods due to policy changes, diurnal cycles, and sudden workload migrations. If the batch size is excessively large, the model may overemphasize the dominant historical regimes within the batch and become less responsive to minority patterns or emerging behaviors, which can degrade prediction quality under distribution shifts.

The overall error-band range remaining relatively stable across batch sizes suggests a certain robustness of the modeling framework, but the best performance still relies on selecting an appropriate batch scale. Batch size is not merely an efficiency parameter in training; it implicitly controls the trade-off between gradient noise and generalization for cloud load prediction. Choosing a suitable batch size allows the model to learn temporal representations that are stable yet adaptive, thereby providing more dependable predictive support for capacity planning, elastic scaling, and resource scheduling decisions in cloud resource management.

6. Conclusion

This paper addresses the challenges of highly dynamic resource loads and complex evolution mechanisms in cloud computing environments. It proposes a dynamic cloud resource load prediction method based on deep time series representation learning. From a unified modeling perspective, the approach enhances the understanding of load evolution patterns. Through systematic representation learning on multivariate historical load sequences, the method effectively integrates temporal information across different time scales under complex operating conditions. It also reduces the impact of transient fluctuations and noise on prediction stability. Comparative results indicate clear advantages in overall error control and prediction consistency. This confirms that a representation-centered modeling paradigm is better suited to the multi-source coupled load generation mechanisms observed in real cloud environments.

From an application perspective, the proposed approach provides practical predictive support for resource management and operational optimization in cloud platforms. More accurate and stable load predictions help establish a more reliable forward-looking relationship between resource supply and actual demand. This reduces resource waste and performance risks caused by delayed or biased predictions. For cloud native systems that rely on elastic scaling, automatic scheduling, and service quality assurance, the method enhances awareness of future operating states. It lays a foundation for more refined and intelligent resource allocation. It also contributes to improved resilience of large-scale cloud platforms under high concurrency and complex load conditions.

In broader application scenarios, the prediction framework based on deep time series representation learning is not limited to a single resource dimension. It also shows potential for extension to multi-resource joint modeling and cross-service load analysis. As cloud system architectures continue to evolve, load behaviors increasingly exhibit cross-component correlations and cross-time scale coupling. This places higher demands on the expressive power and generalization ability of prediction models. The representation learning approach adopted in this work offers a general modeling pathway for such complex temporal processes. It is expected to provide broader value in cloud performance evaluation, capacity planning, and operational state analysis.

For future research, the framework can be further extended to address emerging operating patterns and management requirements in cloud computing environments. On one hand, joint modeling with cloud system structure information, operational constraints, and management policies can be explored to enhance interpretability and decision relevance. On the other hand, the adaptability and stability of the method can be examined in more complex and longer-term operating scenarios. As cloud platforms continue to grow in scale and complexity, more forward-looking and robust load prediction methods will become essential for intelligent cloud resource management. The work presented here offers an exploratory contribution to this direction.

References

[1] J. Gao, H. Wang, and H. Shen, "Machine learning based workload prediction in cloud computing," in Proc. 2020 29th International Conference on Computer Communications and Networks (ICCCN), IEEE, 2020, pp. 1-9.

[2] T. Wang, "Predictive vertical CPU autoscaling in Kubernetes based on time-series forecasting with Holt-Winters exponential smoothing and long short-term memory," 2021.

[3] M. Cioca and I. C. Schuszter, "A system for sustainable usage of computing resources leveraging deep learning predictions," Applied Sciences, vol. 12, no. 17, Art. no. 8411, 2022.

[4] B. Suleiman, M. J. Alibasa, Y. Y. Chang, et al., "Predictive auto-scaling: LSTM-based multi-step cloud workload prediction," in Proc. International Conference on Service-Oriented Computing, Singapore: Springer Nature Singapore, 2023, pp. 5-16.

[5] L. Ruan, H. Guo, Y. Xue, et al., "Towards workload trend time series probabilistic prediction via probabilistic deep learning," in Proc. 18th International Symposium on Spatial and Temporal Data, 2023, pp. 41-50.

[6] H. Yuan and S. Liao, "A time series-based approach to elastic Kubernetes scaling," Electronics, vol. 13, no. 2, Art. no. 285, 2024.

[7] L. Wen, M. Xu, A. N. Toosi, et al., "TempoScale: A cloud workloads prediction approach integrating short-term and long-term information," in Proc. 2024 IEEE 17th International Conference on Cloud Computing (CLOUD), IEEE, 2024, pp. 183-193.

[8] A. Lackinger, A. Morichetta, and S. Dustdar, "Time series predictions for cloud workloads: A comprehensive evaluation," in Proc. 2024 IEEE International Conference on Service-Oriented System Engineering (SOSE), IEEE, 2024, pp. 36-45.

[9] M. Smendowski and P. Nawrocki, "Optimizing multi-time series forecasting for enhanced cloud resource utilization based on machine learning," Knowledge-Based Systems, vol. 304, Art. no. 112489, 2024.

[10] S. Sun, "TrendFormer: Two-stage trend fusion for financial time series forecasting using transformers," 2024.

[11] Y. Sun, "Prior enhanced representation learning and adaptive recognition method for backend anomaly detection under weakly labeled and few-shot conditions," 2024.

[12] Y. Nie, "Financial risk identification using unified multi-source feature learning and structural aggregation," 2024.

[13] K. Zeng, "Transformer-based structure-aware lung cancer image segmentation with multi-scale fusion and boundary-guided prediction," 2024.

[14] S. Y. Huang, "Retrieval-augmented long-text reasoning with semantic conflict detection and cross-paragraph evidence integration," 2024.

[15] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, vol. 9, no. 8, pp. 1735-1780, 1997.

[16] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," IEEE Transactions on Signal Processing, vol. 45, no. 11, pp. 2673-2681, 1997.

[17] A. Vaswani, N. Shazeer, N. Parmar, et al., "Attention is all you need," Advances in Neural Information Processing Systems, vol. 30, 2017.

[18] S. Wang, H. Wu, X. Shi, et al., "TimeMixer: Decomposable multiscale mixing for time series forecasting," arXiv preprint arXiv:2405.14616, 2024.

[19] H. Zhou, S. Zhang, J. Peng, et al., "Informer: Beyond efficient transformer for long sequence time-series forecasting," in Proc. AAAI Conference on Artificial Intelligence, vol. 35, no. 12, 2021, pp. 11106-11115.