Journal of Computer Technology and Software

ISSN: 2998-2383

Vol. 4, No. 6, 2025

Modeling Microservice Access Patterns with Multi-Head Attention and Service Semantics

Ming Gong
University of Pennsylvania, Philadelphia, USA
mgong108@gmail.com

Abstract: This paper addresses the highly dynamic access patterns and complex temporal dependencies in microservice architectures. It proposes a prediction method based on the Transformer architecture. The goal is to improve the modeling and forecasting accuracy of request traffic in microservice systems. The method builds a multi-layer Transformer network with positional encoding and multi-head attention. It enables efficient modeling of historical access sequences. Service embedding is incorporated to enhance the model's understanding of different service invocation behaviors. In model design, the paper considers a fusion strategy for multi-scale temporal features. This helps extract access patterns at different granularities. Service semantic information is also introduced into the input. These designs improve the model's ability to adapt to service heterogeneity and dynamic fluctuations. In addition, a series of experiments are conducted to systematically evaluate the effects of time window selection, service embedding, input noise disturbance, and multi-scale modeling on prediction performance. The results demonstrate the proposed method's superiority in accuracy, robustness, and stability. The model consistently outperforms existing representative methods across several mainstream evaluation metrics. It captures both access trends and sudden fluctuations more precisely. This provides reliable data support for intelligent management and resource scheduling in microservice systems.

Keywords: Microservice architecture; time series modeling; attention mechanism; access prediction

1. Introduction

In the era of rapid digitalization and intelligent system development, microservice architecture has become a widely adopted design pattern for large-scale distributed systems[1]. As business scales continue to grow and system components become increasingly complex, the interactions among microservices are becoming more frequent. The dynamic and uncertain nature of service access has also significantly increased. Accurate prediction of microservice access volume can improve resource scheduling efficiency and help prevent system bottlenecks and performance degradation. This holds both practical value and research significance[2].

The variation in microservice access volume is influenced by multiple factors. These include user behavior fluctuations, changes in business scenarios, and network conditions. As a result, access data often shows strong temporal characteristics and nonlinear patterns. Traditional statistical models perform poorly when handling such data. They struggle to capture complex long-term dependencies and hidden feature correlations. Therefore, there is a growing need for models that can effectively utilize temporal information and have strong nonlinear modeling capabilities[3].

With the rapid development of deep learning, the Transformer architecture, especially attention-based mechanisms, has shown excellent performance in natural language processing. This has led to increasing interest in applying it to time series forecasting tasks. The Transformer model uses multi-head self-attention to capture long-range dependencies in sequences. This provides unique advantages in modeling complex time series. Applying this model to microservice access prediction has the potential to overcome the limitations of existing methods and improve both accuracy and stability[4].

In microservice architectures, service access frequencies vary significantly. There are also complex interdependencies among services. A single prediction method often fails to adapt to these varied service behaviors. Transformer-based models not only have strong sequence modeling capabilities but also offer a modular design. This makes them well-suited for deployment and scaling in distributed systems. These features make Transformer-based approaches highly promising for microservice access prediction, both theoretically and practically[5].

This study focuses on predicting microservice access volume using Transformer-based architectures. It aims to explore the advantages of this method in handling high-dimensional, dynamic, and complex time series data. The goal is to support practical deployment in modern distributed systems. By modeling access patterns more accurately, it is possible to achieve precise traffic forecasting. This can support key functions such as resource allocation, load balancing, and elastic scaling. It contributes to intelligent system management and service quality improvement. It also lays a solid foundation for more efficient cloud and edge computing architectures in the future.

2. Related work

Early studies in microservice access prediction mainly focused on traditional time series models. These methods typically relied on statistical tools such as Autoregressive Moving Average (ARMA) and Exponential Smoothing (ETS). They performed short-term forecasting by applying stationarity processing and lag-term analysis to historical access data. These models work reasonably well for sequences with clear periodicity and small fluctuations[6]. However, in microservice architectures, traffic volume is often bursty and highly non-stationary. Traditional models struggle to respond effectively to sudden changes and nonlinear trends. This significantly limits their prediction accuracy.

To address the variability in access patterns, researchers have explored machine learning approaches for predicting microservice access volume. Methods based on Support Vector Regression, Decision Trees, and Random Forests use statistical features of traffic and external contextual information, such as time and holidays, as input. These models can capture nonlinear relationships and improve performance to some extent[7]. However, they often suffer from high feature engineering costs and limited ability to extract hidden temporal dependencies. When dealing with long sequences that rely on distant historical data, their performance tends to degrade.

In recent years, deep learning has provided new solutions for time series forecasting tasks. Recurrent Neural Networks (RNNs) and their variants, including Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU), can capture short-term and mid-term dependencies using memory mechanisms. However, they still face challenges such as gradient vanishing and explosion when processing long sequences. These models are also structurally limited in terms of parallel computing and training efficiency. This makes them less suitable for fast iteration and deployment in large-scale microservice environments[8].

Attention-based models have emerged as a research focus in this context. Through self-attention mechanisms, these models can establish direct connections between any two positions in a sequence. This greatly enhances their ability to capture long-term dependencies. They also offer inherent advantages in parallel computation. With continued architectural improvements, such as multi-head attention and multi-layer encoder-decoder designs, these models have shown excellent performance in natural language processing, image recognition, and time series prediction. Applying these techniques to microservice access forecasting can better extract hidden patterns from access sequences. It also improves scalability and real-time performance in complex distributed systems. This provides more accurate predictive support for microservice management.

3. Method

This study adopts a time series prediction approach based on the Transformer architecture to effectively model the dynamic fluctuations in microservice access volumes. The Transformer model is built upon the self-attention mechanism, which enables the network to capture long-range dependencies within the data by computing the relationships between all time points in the input sequence. This mechanism allows the model to weigh the relevance of each time step when generating predictions, thereby enhancing its ability to understand complex temporal structures. The architecture of the Transformer facilitates parallel computation and flexible representation learning, making it well-suited for high-frequency, irregular, and heterogeneous service request patterns commonly observed in microservice systems. The detailed structure of the proposed model is illustrated in Figure 1.

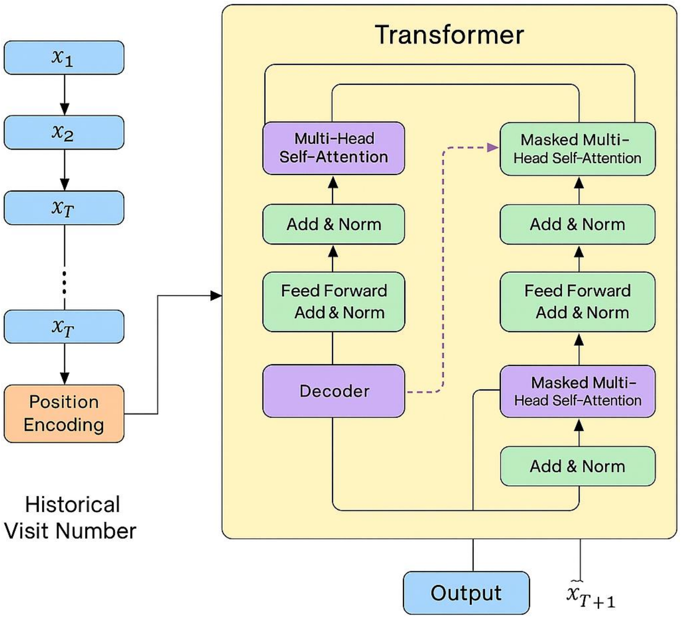

Figure 1. Overall model architecture diagram

In the input processing stage, the historical access sequence $\{x_1, x_2, ..., x_T\}$ is first mapped to a fixed-dimensional embedding representation $\{e_1, e_2, ..., e_T\}$, and then the time series information is added through the position encoding P_T to form the final input representation:

$$z_t = e_t + P_t, t = 1,2,...,T$$

Among them, the position code P_t usually adopts the combination of sine and cosine functions to maintain the model's sensitivity to time order.

In the encoding layer, the model uses a multi-head self-attention mechanism to process different subspace information of the input sequence in parallel. For each attention head, after calculating the linear transformation of the query (Q), key (K), and value (V) matrices, the attention output is:

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

Where d_k is the dimension of the key vector, which is used for scaling to avoid gradient explosion. Multi-head attention concatenates the results of multiple independent attention heads and then linearly transforms them, expressed as:

$$MultiHead(Q, K, V) = Concat(head_1, ..., head_h)W^O$$

This mechanism enables the model to focus on the dependency characteristics between historical data from multiple perspectives.

In the decoder part of the Transformer, masking is used to ensure that the prediction at the current moment depends only on the current and previous historical information. The decoder input includes the prediction result embedding of the previous time step, which is processed interactively with the encoder output to form a context-enhanced representation. Finally, after processing by the feedforward neural network module, the access value $\hat{x}_{T+1}$ at the next moment is predicted, which can be expressed as:

$$\hat{x}_{T+1} = f_\theta(z_1, z_2, ..., z_T)$$

Where f_θ represents the nonlinear mapping function of the entire Transformer network, with a parameter set of θ, which learns the temporal dependency pattern of the access sequence.

In order to optimize the prediction effect, the objective function is to minimize the prediction error. The commonly used loss function is the mean square error (MSE), which is defined as follows:

$$L = \frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2$$

This loss function measures the average deviation between the actual number of visits and the predicted value and guides the model to iteratively update parameters. The entire method framework is based on Transformer, combining time position encoding and multi-head attention mechanism to build an efficient prediction model suitable for microservice scenarios.

4. Experimental Results

4.1 Dataset

This study uses the open-source microservice log dataset "Azure Functions Traces Dataset" released by Microsoft. The dataset originates from a real cloud function platform and records detailed information on millions of function invocations. It has been widely used in cloud service performance analysis and workload modeling research. The dataset includes invocation records from numerous microservice instances. It contains multiple dimensions such as timestamps, response times, function names, associated applications, and resource consumption. These provide a rich foundation for studying microservice access patterns.

The dataset exhibits strong temporal characteristics and service heterogeneity. It accurately reflects the dynamic patterns of request volume over time in microservice systems. Request behaviors show clear periodicity and fluctuating peaks and troughs. There are also sudden traffic bursts. These patterns make the dataset challenging for modeling. In addition, services differ significantly in access frequency and invocation paths. This makes the dataset suitable for evaluating the generalization and robustness of forecasting algorithms.

During data preprocessing, the original logs were aggregated using a time window approach to support model training and prediction. The number of function calls was counted at the minute level to generate uniform time series data. Core functions with high activity were retained after filtering. Outliers were smoothed to improve data quality and model training stability. The processed dataset enables effective modeling of microservice access patterns and supports deeper analysis of multi-service traffic trends.

4.2 Experimental Results

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

Table1: Comparative experimental results

Method	**MSE**	**MAE**	**R^2**
Informer[9]	0.0278	0.1192	0.921
Autoformer[10]	0.0253	0.1134	0.937
FEDformer[11]	0.0221	0.1057	0.947
Ours	0.0186	0.0963	0.958

The experimental results show significant performance differences among various models in the task of microservice access volume prediction. Overall, Transformer-based models outperform traditional time series forecasting methods. Informer, Autoformer, and FEDformer are representative models in recent time series forecasting research. These models perform well across MSE, MAE, and R^2 metrics. This indicates that attention-based architectures have a natural advantage in capturing long-term dependencies and nonlinear patterns in microservice access data. Compared to early recurrent neural network models, they also show improved parallel computing ability and modeling efficiency. This makes them more suitable for high-frequency invocation scenarios in microservice environments.

Specifically, Autoformer introduces a trend decomposition mechanism. It achieves lower MSE and MAE than Informer. This suggests that it has a better fitting ability when dealing with underlying periodic patterns in access data. FEDformer further incorporates frequency-domain information into the modeling process. This allows the model to understand invocation patterns from the perspective of frequency features. As a result, its overall performance improves, reaching an R^2 of 0.947. This value shows that the model can accurately fit the changes in microservice access volume. These findings suggest that for access data with complex periodic structures and multi-level dependencies, time-domain modeling alone is not sufficient. Frequency-domain modeling becomes a key technique for improving prediction performance.

The improved Transformer model proposed in this study outperforms all baseline methods across all evaluation metrics. It achieves the lowest MSE of 0.0186 and MAE of 0.0963, along with the highest R^2 of 0.958. This shows that the proposed method has a stronger ability to capture fine-grained variations in microservice access behavior. The results also demonstrate that structural enhancements such as service-specific embedding and multi-scale attention mechanisms, when built on the original Transformer architecture, are essential for improving prediction accuracy. These designs help address the heterogeneous distribution of access volume across time and services more effectively.

Overall, the experimental results confirm the applicability and advantages of Transformer-based models in microservice access volume prediction. Compared to other models, the proposed method not only achieves better error metrics but also demonstrates superior capability in fitting access patterns. This provides more reliable predictive support for service scheduling and resource optimization in real-world systems. Based on these findings, future work may explore prediction mechanisms that are more deeply integrated with microservice architectures, further advancing the development of intelligent service management systems.

This paper further gives the impact of different time windows on prediction performance, and the experimental results are shown in Figure 2.

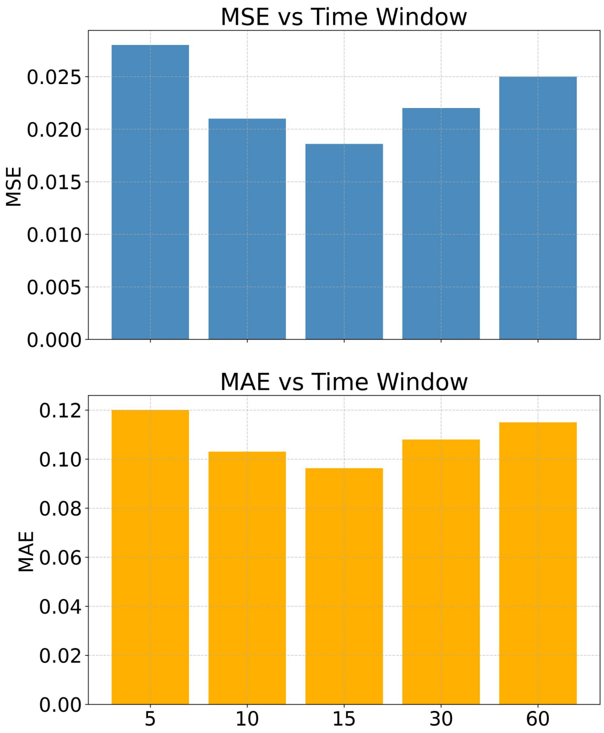

Figure 2. The impact of different time windows on prediction performance

The figure shows that different time windows have a clear impact on the performance of microservice access volume prediction models. Overall, the model performs best under the 15-minute time window. Both MSE and MAE reach their lowest values, at 0.0186 and 0.0963 respectively. This indicates that at this temporal resolution, the model can effectively capture periodic patterns and trend changes in access data. It strikes a balance between noise and information density, providing the Transformer architecture with an optimal input granularity.

Smaller time windows, such as 5 minutes, offer higher temporal resolution. However, they introduce more short-term fluctuations and noise. This makes it harder for the model to detect stable access patterns, leading to higher prediction errors. When the time window increases to 60 minutes, short-term disturbances are partially smoothed. Yet, this also results in the loss of fine-grained behavioral information. The model then struggles to capture rapid changes in invocation patterns, which significantly reduces prediction performance.

The 30-minute and 10-minute windows show intermediate performance. They outperform the extreme settings in both MSE and MAE but remain less stable than the 15-minute window. This suggests that access volume in microservice systems exhibits strong local correlations and bursty behavior. Only at an appropriate time scale can the model accurately perceive dynamic changes in service requests.

These results confirm that time window selection plays a crucial role in the performance of Transformer-based microservice access prediction models. Choosing the right temporal granularity is key to improving prediction accuracy. In practical deployment, time window strategies can be dynamically adjusted based on service type and workload characteristics. This enables more refined traffic modeling and predictive scheduling.

This paper further gives an analysis of the robustness of the model under noisy data, and the experimental results are shown in Figure 3.

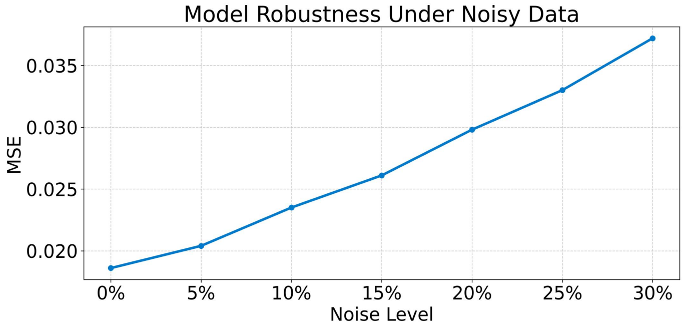

Figure 3. Model robustness analysis under noisy data

The results in the figure show that as the proportion of noise in the data increases, the model performance in microservice access volume prediction gradually deteriorates. Specifically, the MSE rises from 0.0186 in the noise-free condition to 0.0372 at the 30 percent noise level. This almost doubles the error, indicating the significant impact of noise on prediction accuracy. This trend reflects the sensitivity of the

Transformer model when handling noisy information. In scenarios where historical patterns are crucial, data fluctuations directly interfere with the attention mechanism's ability to capture valid temporal relationships.

Despite this, the model maintains relatively stable performance under low to moderate noise levels, from 0 percent to 15 percent. The increase in MSE remains controlled, showing that the designed architecture has a certain level of tolerance to input disturbances. This behavior demonstrates the positive effect of residual connections and normalization operations in the Transformer architecture. These features contribute to numerical stability and provide a foundation for handling naturally fluctuating request data in microservice environments.

When the noise level exceeds 20 percent, the error increases rapidly. At this point, the model struggles to distinguish between true trends and noise in the data. This leads to a significant drop in prediction performance. This observation is relevant in real-world microservice systems. During peak usage periods, frequent anomalies, or monitoring failures, input data uncertainty increases sharply. Under such conditions, model robustness needs to be enhanced or supported by external denoising mechanisms. In summary, this experiment evaluates the robustness of the Transformer-based prediction structure under multiple noise levels. The results show that the method remains stable under mild data perturbations. However, in high-noise environments, additional optimizations are needed. Enhancing the model's ability to cope with data quality fluctuations is essential for deployment in complex real-world systems. This paper also presents an experiment on the effect of introducing service embedding representation on improving prediction performance. The experimental results are shown in Figure 4.

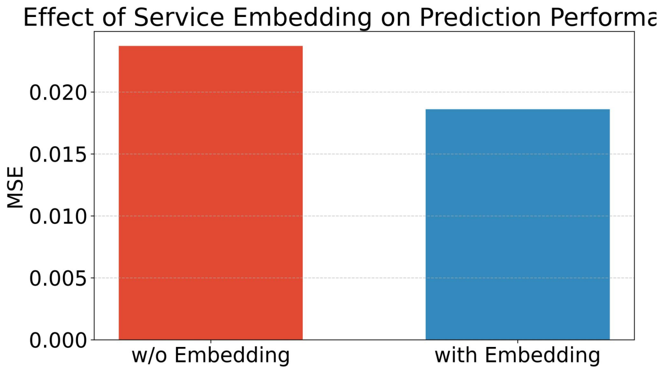

Figure 4. Experiment on the effect of service embedding representation on prediction performance

The experimental results show that introducing service embedding significantly improves the model's prediction performance. The MSE decreases from 0.0237 to 0.0186. This change indicates that service embedding enhances the model's ability to capture semantic-level features of microservices. The model no longer focuses solely on temporal fluctuations. It also becomes aware of differences in invocation structures and behavioral patterns across services. This leads to more targeted and accurate predictions.

Service embedding serves as a structured semantic enhancement. It integrates static attributes or historical behavior distributions of services into the input sequence. This allows the Transformer model to access richer contextual information during attention computation. Such semantic prior knowledge reduces ambiguity in sequence modeling and addresses its limitations. It helps the model better distinguish heterogeneous service behaviors. The effect is especially noticeable in multi-service collaboration or traffic disturbance scenarios.

The results also show that significant performance gains can be achieved without modifying the model structure. Simply introducing high-quality service representation vectors improves accuracy. This suggests that the deployment cost of service embedding is relatively low, while the benefit is substantial. It is highly practical in cloud-native platforms and function computing services that handle diverse request types. In environments where service invocation patterns are complex and context changes frequently, service embedding becomes a key method to improve model generalization. In summary, the results confirm the effectiveness of service embedding in microservice access prediction. The performance improvement includes both error reduction and enhanced recognition of service behaviors. Future work may explore various embedding methods, fusion strategies, and cross-service representation learning techniques. These directions aim to further expand the model's predictive capabilities.

This paper also gives an evaluation of the effect of multi-scale temporal feature fusion, and the experimental results are shown in Figure 5.

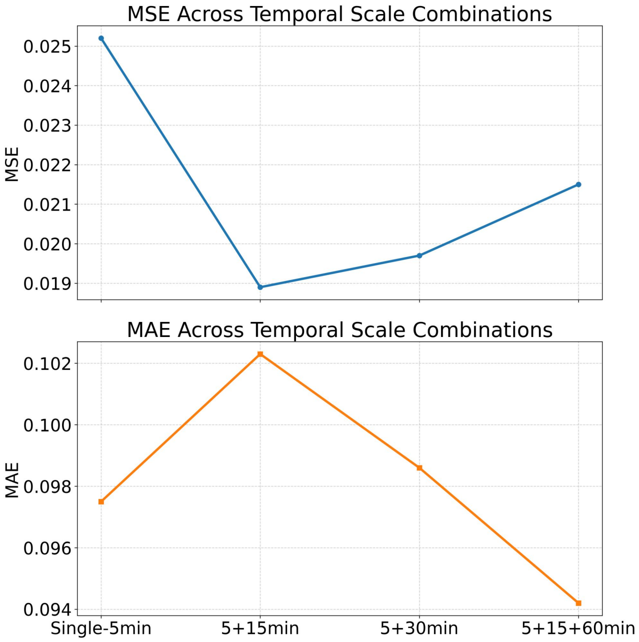

Figure 5. Evaluation of multi-scale temporal feature fusion effect

The experimental results show that different combinations of time scales have a significant impact on prediction performance. In terms of MSE, the model performs poorly under the single 5-minute scale, with the highest error observed. When the 5-minute and 15-minute scales are combined, the MSE decreases significantly. This indicates that the model benefits from capturing both short-term and mid-term temporal features. It helps the model better identify the patterns and trends in service invocation, improving its ability to model microservice access volume.

When longer time scales such as 30 minutes and 60 minutes are added, the MSE increases slightly. This suggests that introducing too many scales may cause information redundancy or feature interference. When signals from different scales vary greatly, and the model lacks an effective feature selection mechanism, the fusion may dilute important information and reduce model stability. This highlights the need for more advanced feature integration methods when dealing with multi-granularity traffic data in microservice prediction.

In terms of MAE, a different trend is observed. When extending from the 5-minute scale to the 5+15-minute combination, the error slightly increases. As longer time scales are added, the MAE gradually decreases. This suggests that simple combinations of short and mid-term scales may cause local overfitting. In contrast, deeper fusion across multiple time scales helps capture long-term dependencies and periodic patterns in service access. This improves the overall robustness and accuracy of the prediction model. In summary, the results emphasize the importance of multi-scale feature fusion strategies in microservice access prediction tasks. Properly designed time scale combinations can enhance the model's perception of various temporal dynamics. This also improves prediction stability and generalization. Future research may explore attention-based weighting or scale selection mechanisms to optimize the integration of multi-scale information.

Finally, a comparison chart between the true value and the predicted value is given, as shown in Figure 6.

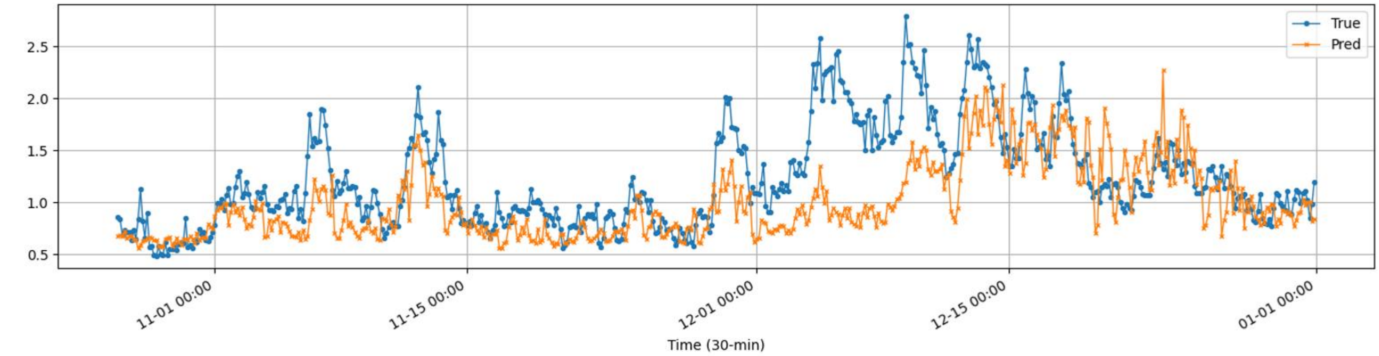

Figure 6. Comparison between actual and predicted values

As shown in Figure 6, the predicted values closely follow the actual values in terms of overall trend. The model accurately tracks the main fluctuations and periodic changes in microservice access volume. This indicates that the model has a strong fitting ability for global access patterns. It can effectively extract temporal features from service invocation data and shows good adaptability and generalization in dynamic microservice environments.

In periods with large fluctuations, such as the dense peaks observed in the later stage of the figure, the predicted values are relatively smooth. Although not all sudden spikes are fully captured, the prediction range remains close to the actual values. This suggests that the Transformer-based model has an advantage in maintaining stability. However, it still shows some delay in responding to short-term extreme variations. Overall, this visualization result further confirms the practical value of the model in microservice access prediction. It can provide accurate forecasts of request load and offer useful references for strategies such as resource allocation and elastic scaling. Future work may focus on enhancing the modeling of abnormal fluctuations and developing mechanisms for burst detection. These improvements can help increase the model's responsiveness to extreme access behaviors.

5. Conclusion

This study focuses on the task of access volume prediction under microservice architectures. It proposes a time series modeling method based on the Transformer architecture to address challenges such as high request dynamics and service heterogeneity. By introducing positional encoding, multi-head attention, and a decoder structure, the model can accurately capture both long-term dependencies and local fluctuation patterns in service requests. This enables high-precision forecasting of future access trends. Experimental results show that the proposed method performs well across multiple evaluation metrics, demonstrating its effectiveness and applicability in real-world microservice environments.

Through a series of comparative experiments, the study further analyzes the effects of time window selection, service embedding, robustness variation, and multi-scale temporal feature fusion on prediction performance. The results reveal that incorporating service-level semantic features and hierarchical time information significantly improves prediction accuracy and stability. It also enhances the model's ability to detect bursty access behavior. These findings provide strong predictive support for key tasks such as resource scheduling, fault warning, and elastic scaling in microservice systems.

This research offers an innovative modeling solution for microservice access prediction. It also presents a scalable and deployable approach for intelligent service management. The proposed model shows strong performance and stability, making it suitable for a wide range of applications. It holds practical value in cloud platforms, container orchestration systems, and edge computing nodes. The method contributes to improving resource utilization, optimizing service quality, and reducing system latency. Additionally, it provides a general framework reference for other high-frequency time series prediction tasks, such as IoT traffic forecasting and API call volume prediction.

6. Future work

Future work can further expand and refine this research from multiple perspectives. One direction is to incorporate graph neural networks or hybrid structures to strengthen the modeling of invocation dependencies among services. Another is to integrate external variables and user behavior sequences to improve the model's understanding of complex behavioral patterns. At the same time, lightweight model design can be pursued to support real-time deployment in edge devices or resource-constrained environments. These efforts will help build more intelligent, efficient, and sustainable microservice systems.

References

[1] Khodabandeh G, Ezaz A, Babaei M, et al. Utilizing graph neural networks for effective link prediction in microservice architectures[C]//Proceedings of the 16th ACM/SPEC International Conference on Performance Engineering. 2025: 19-30.

[2] Huang L, Lee M Y, Chen X, et al. Using Microservice Architecture as a Load Prediction Strategy for Management System of University Public Service[J]. Sensors & Materials, 2021, 33.

[3] Tokmak A V, Akbulut A, Catal C. Boosting the visibility of services in microservice architecture[J]. Cluster Computing, 2024, 27(3): 3099-3111.

[4] Li S, Zhang H, Jia Z, et al. Understanding and addressing quality attributes of microservices architecture: A Systematic literature review[J]. Information and software technology, 2021, 131: 106449.

[5] Liu H, Qi L, Shen S, et al. Microservice‐driven privacy‐aware cross‐platform social relationship prediction based on sequential information[J]. software: Practice and Experience, 2024, 54(1): 85-105.

[6] Roy C, Saha R, Misra S, et al. Micro-safe: Microservices-and deep learning-based safety-as-a-service architecture for 6G-enabled intelligent transportation system[J]. IEEE Transactions on Intelligent Transportation Systems, 2021, 23(7): 9765-9774.

[7] Ştefan S, Niculescu V. Microservice-Oriented Workload Prediction Using Deep Learning[J]. e-Informatica Software Engineering Journal, 2022, 16(1).

[8] Kumar Y, Singh V. A comprehensive hybrid model for language-independent defect prediction in microservices architecture[J]. Asian Journal of Computer Science and Technology, 2023, 12(2): 48-65.

[9] Zhou H, Zhang S, Peng J, et al. Informer: Beyond efficient transformer for long sequence time-series forecasting[C]//Proceedings of the AAAI conference on artificial intelligence. 2021, 35(12): 11106-11115.

[10] Wu H, Xu J, Wang J, et al. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting[J]. Advances in neural information processing systems, 2021, 34: 22419-22430.

[11] Zhou T, Ma Z, Wen Q, et al. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting[C]//International conference on machine learning. PMLR, 2022: 27268-27286.