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

Capturing Traffic Propagation and Topological Interactions in Cloud Networks via Graph Neural Networks

Xinxin Huang
Northeastern University, Boston, USA
serenade0018@gmail.com

Abstract: Network traffic in cloud computing environments is jointly influenced by multi-tenant concurrency, service dependency relationships, and dynamic scheduling policies. It therefore exhibits strong volatility, temporal non-stationarity, and structural coupling, which pose significant challenges for accurate prediction. To address the limitation of traditional time series methods in capturing inter-node correlations and traffic propagation mechanisms, this study proposes a graph neural network-based approach for cloud traffic fluctuation prediction from a structural modeling perspective. The cloud network is abstracted as a time-evolving graph, where node-level traffic observations are treated as graph signals. Spatial dependencies among communication entities are modeled through graph structures, while temporal dynamics of traffic evolution are learned via sequence encoding mechanisms. This enables a unified representation of spatiotemporal features within a single framework. Under consistent data settings and evaluation protocols, the proposed method is systematically compared with several representative forecasting models. The results demonstrate clear advantages in error control and prediction stability. The method is more effective in handling traffic fluctuations caused by topological interactions and workload migration in complex cloud scenarios. These findings indicate that structure-aware spatiotemporal modeling can substantially improve the reliability of cloud traffic prediction and provide a more robust foundation for network resource management and operational state analysis.

Keywords: Cloud computing traffic prediction; graph neural networks; spatiotemporal modeling; network load fluctuations

1. Introduction

With the widespread deployment of cloud computing infrastructure in data centers, edge computing environments, and cloud native applications, network traffic has become a critical resource for ensuring platform stability and service quality[1]. Traffic in cloud environments is not only large in scale, but also exhibits strong burstiness, periodicity, and randomness. Its dynamics are closely coupled with business workloads, user behavior, service topology, and scheduling policies. Under multi-tenant and large-scale distributed architectures, traffic fluctuations can easily trigger cascading congestion, amplified service latency, and imbalanced resource utilization. These effects can significantly degrade system reliability and user experience. Therefore, accurate modeling and proactive prediction of cloud traffic fluctuations are fundamental to fine-grained network resource management, load-aware scheduling, and intelligent operations. They also represent a key prerequisite for the transition of cloud platforms from reactive response to proactive control[2].

Despite this importance, the generation mechanism of cloud traffic is highly complex. Its evolution is influenced not only by historical temporal dependencies but also by the underlying structural relationships of the cloud system. Virtual machines, containers, microservices, switches, and communication links form a dynamically evolving network topology. Communication patterns across nodes vary substantially and continue to change with service scaling, migration, and scheduling adjustments. Traditional time series-based or statistical methods often treat traffic as independent sequences. Such approaches fail to explicitly capture structural dependencies and interactive effects among nodes. As a result, their modeling capability is limited in scenarios involving cross-node coupling, local anomaly propagation, or global traffic redistribution. The separation between structural information and temporal features has thus become a major bottleneck in improving the accuracy and stability of cloud traffic prediction[3].

Against this background, it is essential to revisit cloud traffic prediction from a system structural perspective. A cloud platform is inherently a complex network composed of numerous computing and communication entities. Traffic fluctuations do not arise in isolation. They propagate and accumulate along service invocation relationships, data transmission paths, and resource dependencies. Modeling cloud traffic as dynamic signals on graph structures enables a unified representation of node attributes, topological relationships, and temporal evolution. This perspective aligns more closely with the actual operating mechanisms of cloud systems. By incorporating graph-based modeling principles, prediction processes can explicitly account for node roles, local structural constraints, and the influence of global topology on the propagation of traffic fluctuations. This provides a more systematic and interpretable description of complex traffic behaviors.

Furthermore, graph-based learning paradigms offer a natural representation framework for cloud traffic prediction. They allow dependencies to be modeled jointly across spatial and temporal dimensions. Such an approach captures not only the intrinsic temporal patterns of traffic at individual nodes but also the strength of interactions among nodes and their dynamic adjustments under varying system states. Common phenomena in cloud environments include cross-service traffic correlation, influence diffusion from high-load nodes, and global fluctuations triggered by local anomalies. Graph-structured modeling reveals these relationships from a holistic perspective. It also helps mitigate uncertainty caused by noise and non-stationarity. As a result, prediction robustness and consistency under complex operating conditions can be effectively improved[4].

From an application perspective, research on graph neural network-based prediction algorithms for cloud traffic fluctuations has substantial practical value and long-term significance. More accurate traffic forecasts provide reliable support for bandwidth allocation, load balancing, and elastic resource scaling. This reduces the risks of over-provisioning and resource waste, and improves overall platform efficiency. In addition, in-depth modeling of traffic dynamics enables early identification of potential congestion risks and abnormal evolution trends. This supports intelligent operations, fault warning, and service quality assurance. More importantly, a structure-aware prediction paradigm can promote the shift of cloud management from experience-driven and rule-based approaches toward data-driven and model-driven strategies. This lays a theoretical and methodological foundation for building adaptive and self-optimizing intelligent cloud infrastructures.

2. Background

Cloud traffic fluctuation prediction is commonly built on a joint understanding of multi-source monitoring data and system structural information. In real cloud platforms, traffic sequences are collected from multiple observation levels, including host network interfaces, virtual switches, container networks, and service invocation links[5]. Their statistical characteristics often exhibit both long-term and short-term dependencies, bursty peaks, and strong non-stationarity. At the same time, traffic does not evolve as an isolated time series signal. It is embedded in a communication graph composed of nodes and links. Different tenants, services, and transmission paths are connected through heterogeneous dependency and propagation relationships.

These relationships continuously change due to scaling operations, service migration, routing updates, and load scheduling. As a result, relying solely on traditional time series models or local feature engineering is often insufficient to consistently capture cross-node coupling, fluctuation propagation, and structural constraints. This leads to amplified prediction errors, limited generalization capability, and weak sensitivity to abnormal fluctuations in complex cloud scenarios.

To address these challenges, recent studies have gradually shifted toward structure-aware spatiotemporal modeling paradigms. In this setting, the cloud network is abstracted as a graph, and traffic observations at each node are treated as graph signals. Dependencies among nodes are learned in the spatial dimension, while evolution patterns are captured in the temporal dimension. The core idea of this line of work is to integrate topological priors with data-driven representations within a unified framework. Graph-based modeling is used to capture service dependencies, link connectivity, and interaction strength. Temporal modeling is applied to describe periodicity and burstiness[6,7]. Dynamic graphs or adaptive adjacency mechanisms are further introduced to accommodate structural changes over time in cloud environments. Beyond improving prediction accuracy and stability, structured representations also provide more interpretable support for resource scheduling and risk warning. This enables model outputs to align more closely with management objectives such as bandwidth planning, congestion control, and elastic scaling. It therefore establishes a methodological foundation for intelligent network management in the cloud native era.

3. Model Design

3.1 Overall Framework Description

This paper's method starts from the overall structure of cloud computing networks, abstracting the cloud environment into a graph structure that evolves to uniformly characterize the temporal dependence of traffic fluctuations and the structural relationships between nodes. Let the cloud network at time t be represented as graph $G(V, E_t)$, where V represents the set of computing or communication nodes, and E_t represents the communication relationships between nodes at that time. Each node at time t corresponds to a traffic observation vector $x_t \in R^d$, where d is the traffic feature dimension. The overall modeling goal is to learn a mapping function that can predict the node traffic state at future times based on graph signals from multiple historical time steps. Its form can be uniformly expressed as:

$$x_{t+1} = f(x_t, \ldots, x_1)$$

Where x_{t+1} represents the predicted flow rate at the next time step. This modeling approach emphasizes the joint characterization of temporal evolution and structural dependence, providing a unified representation basis for subsequent module design. This paper also presents the overall model architecture, as shown in Figure 1.

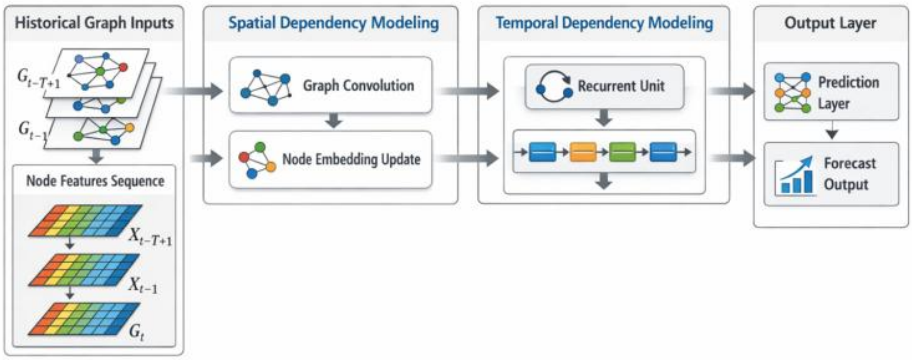

Figure 1. Overall model architecture

3.2 Graph-Based Spatial Dependency Modeling

To characterize the spatial dependencies between cloud computing nodes, this paper employs a graph-based feature propagation mechanism to update the node traffic representation. Given the adjacency matrix A_t of the graph, its normalized form is first constructed:

$$A_t = D_t^{-\frac{1}{2}}(A_t + I)D_t^{-\frac{1}{2}}$$

Where I is the identity matrix and $D_t^{-\frac{1}{2}}$ is the degree matrix. Based on this structure, the feature update process of a node in the spatial dimension can be represented as:

$$H_t = \sigma(\tilde{A}_t X_t W_s)$$

Where X_t is the input feature matrix of the node at time t, W_s is the learnable parameter, and $\sigma(\bullet)$ represents the nonlinear mapping function. This process can aggregate the traffic states of adjacent nodes, enabling the model to explicitly perceive the propagation and coupling characteristics of traffic in the network structure at the representation level.

3.3 Temporal Dependency Encoding

After completing the spatial structural modeling, to further capture the dynamic patterns of flow changes over time, this paper introduces a time-series encoding mechanism to model the node representation. For each node, its spatial feature representation within a continuous time window can be denoted as $\{h_{t-T+1}, \ldots, h_t\}$. Time-dependent modeling is implemented recursively, and its state update process is defined as:

$$s_t = \Phi(h_t, s_{t-1})$$

Where s_t represents the time-encoded state, and $\Phi(\bullet)$ represents the time update function. To reduce the impact of noise and non-stationary fluctuations, the time-dimensional representation is further transformed into an intermediate prediction representation through linear mapping:

$$z_t = s_t W_t + b_t$$

W_t and b_t are learnable parameters. This design enables the model to form a continuous characterization of the flow evolution trend over time, without relying on instantaneous observations at a single moment.

3.4 Prediction Objective and Optimization Formulation

Based on the fusion of spatial and temporal features, the final flow prediction is accomplished by the output mapping function, which is defined as follows:

$$x_{t+1} = z_t W_o + b_o$$

Where W_o and b_o are the output layer parameters. The model training process aims to minimize the difference between the predicted values and the actual traffic flow. The overall optimization objective function can be expressed as:

$$L = \frac{1}{N}\sum_{i=1}^{N} \| \hat{x}_{i+1}^{(i)} - x_{i+1}^{(i)} \|_2^2$$

Where N represents the number of samples. By optimizing the objective function end-to-end, the model can collaboratively learn graph structure dependencies and temporal evolution patterns within a unified framework, thereby forming a systematic modeling capability for cloud computing traffic fluctuations and providing stable representation support for subsequent resource management and predictive decision-making.

4. Implementation

4.1 Dataset

This paper selects the MAWI Working Group Traffic Archive (MAWI) as the data source for cloud computing traffic fluctuation prediction. This dataset provides long-term, publicly available traffic packet captures and corresponding timestamps from real network links, covering typical fluctuation patterns such as mixed applications, burst peaks, and daily cycles. It can effectively simulate the traffic fluctuation characteristics of cloud platform ingress/egress and east-west communication under high concurrency conditions. Because the data comes from a real operating network, its distribution exhibits significant non-stationarity and long-tail bursts, making it suitable for verifying the ability of spatiotemporal modeling methods for fluctuation prediction to characterize complex dynamics.

In data processing, the original packet captures are aggregated into time-series samples at a fixed time granularity, and "traffic intensity and structure statistics" are used as the basic components of node features. Specifically, within each time window, the number of bytes, number of packets, average packet length, uplink/downlink ratio, and several simple flow-level statistics are statistically analyzed to form the feature vector of the node at time t. Simultaneously, outliers are truncated or logarithmically transformed to mitigate the interference of peak values on learning, and each feature dimension is standardized to ensure that inputs under different dimensions are comparable on the same scale. The resulting multivariate sequence, composed of continuous time windows, is input into the model as a graph signal.

To correspond with the spatial modeling of graph neural networks, this paper further constructs a dynamic graph structure from the aggregated traffic: communication entities (e.g., IP prefixes/host sets/traffic aggregation points) are treated as nodes, and pairs of entities with significant communication associations within a time window form edges. The edge weights are determined by the communication intensity (e.g., byte count or traffic percentage) within that time window, thus obtaining adjacency relationships updated over time. Based on this graph sequence, supervised samples are generated using a historical window approach. Training/validation/testing are divided into time sequences to avoid information leakage, ensuring that the prediction task strictly adheres to the setting of "predicting the future using only the past." This constructed data preserves the temporal fluctuations of real traffic while explicitly introducing cross-node structural coupling, providing a complete and reproducible experimental foundation for subsequent spatiotemporal joint modeling.

4.2 Experimental setup

All experiments were conducted on a single machine with a single GPU configuration. The operating system was Ubuntu 22.04 LTS. The CPU was an Intel Xeon Gold 6248R, and the system was equipped with 128 GB of memory. The GPU was an NVIDIA RTX 4090 with 24 GB of memory. The software environment included CUDA 12.1, cuDNN 9.x, Python 3.10.13, and PyTorch 2.1.2. Data loading and preprocessing were implemented using NumPy 1.26, Pandas 2.1, and Scikit learn 1.3. Graph structure computation was performed with PyTorch Geometric 2.4. Training logs and metrics were recorded using TensorBoard 2.14. A fixed random seed of 2025 was used for all experiments. Deterministic training was enabled, and non-deterministic operators were disabled to eliminate stochastic variation and ensure reproducibility across runs.

The hyperparameter configuration was as follows. The historical window length was set to 12, and the prediction horizon was 1, with a temporal resolution of 5 minutes. The node feature dimension was 16. The graph convolution module consisted of two layers with a hidden dimension of 128. The activation function was ReLU, and the dropout rate was 0.2. The temporal modeling module adopted a GRU with one layer and a hidden dimension of 128. The output head was implemented as a two-layer fully connected network with dimensions 128 to 64 to 1. The optimizer was AdamW with an initial learning rate of 0.0003 and a weight decay of 0.0001. The batch size was 64, and the number of training epochs was 100. Gradient clipping was applied with a threshold of 1.0. A cosine annealing learning rate scheduler was used, with a minimum learning rate of 0.00001. Linear warmup was applied during the first five epochs. The dataset was split in

chronological order into training, validation, and test sets with proportions of 70 percent, 10 percent, and 20 percent. Normalization statistics were computed using only the training set and then applied to the validation and test sets to prevent information leakage.

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	RMSE	MAPE
Transformer[8]	0.784	0.612	0.885	6.91
Informer[9]	0.751	0.597	0.867	6.54
TimeMixer[10]	0.703	0.563	0.839	6.21
ITransformer[11]	0.668	0.547	0.812	5.87
Easytime[12]	0.641	0.529	0.798	5.73
Ours	0.588	0.491	0.754	5.12

A comprehensive comparison shows that the proposed method achieves consistent advantages across all four error metrics. This indicates a more stable fitting of traffic fluctuations in cloud computing scenarios. The model maintains lower prediction bias under conditions where bursty peaks and periodic variations coexist. Compared with baseline approaches that mainly rely on global sequence modeling, the results demonstrate stronger adaptability to non-stationarity and multi-scale variation. They also better reflect the rapid switching behavior of cloud traffic driven by workload dynamics and scheduling perturbations.

When compared with several representative long sequence forecasting models, the improvement is not limited to a single metric. Gains are observed simultaneously in both absolute and relative error measures. This implies that the model can more accurately track overall traffic levels and better capture variation amplitudes across different load regimes. For cloud resource management, such consistent error reduction is particularly valuable. It reduces the risk of overestimation under low load conditions and underdetection under high load conditions. This directly enhances the reliability of decisions related to bandwidth reservation, congestion warning, and elastic scaling.

From a methodological perspective, these improvements are commonly attributed to stronger joint modeling of structural and temporal dependencies. Traffic in cloud networks does not emerge in isolation. Communication correlations among nodes cause fluctuations to propagate and couple across the topology. A purely time series-based perspective often overlooks error accumulation induced by cross-node interactions. The current results indicate that the model can better exploit upstream and downstream relationships and local interaction information to constrain predictions. As a result, stable prediction quality is preserved even under topology changes, hotspot migration, or fluctuations in service invocation chains.

In addition, lower relative error values reflect improved adaptation to fluctuations at different scales. This property is particularly important in cloud environments with long-tailed traffic distributions and pronounced peak valley gaps. In such cases, prediction stability directly affects the safety margin and cost efficiency of resource scheduling. Overall, these results support the spatiotemporal modeling perspective emphasized in this study for cloud traffic fluctuation prediction. By learning representations that are more aligned with system operational structures, prediction reliability is enhanced. This provides a more robust foundation for capacity planning, prior anomaly identification, and service quality assurance in intelligent cloud operations.

The prediction step size determines the future time span that the model needs to cover in advance, and it is also the key setting that most directly reflects the difficulty of the task in traffic fluctuation prediction. By changing the prediction step size while keeping other configurations unchanged, we can observe the error variation of the model under different look-ahead spans, thereby assessing its adaptability to long-term uncertainties. The experimental results are shown in Figure 2.

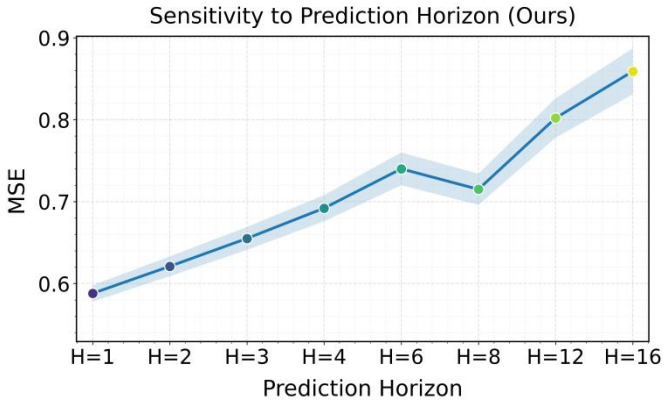

Figure 2. Sensitivity experiment of prediction step size to MSE

As the prediction horizon increases, the curves show an overall upward trend. This indicates that uncertainty in cloud traffic fluctuations grows significantly at longer forecast horizons. The model must simultaneously handle periodic variations, bursty peaks, and non-stationary induced drift. This behavior is consistent with real cloud networks, where scheduling changes, workload hotspot migration, and multi-tenant competition cause rapid shifts in traffic patterns. It also suggests that long-horizon forecasting relies more on abstract modeling of evolutionary dynamics rather than short-term inertia.

The curves do not increase in a strictly monotonic or smooth manner. Instead, stage-wise fluctuations are observed. This reflects threshold effects and structural differences in how prediction horizons influence error. In cloud scenarios, such behavior often arises from mixed driving forces across multiple time scales. Short-cycle request aggregation interacts with longer cycle business rhythms. Topological interactions further introduce propagation delays and strengthened coupling. As a result, certain horizons tend to amplify errors, while others align more closely with dominant cycles and remain relatively stable.

From a methodological perspective, the sensitivity results indicate that the model can effectively exploit spatial dependencies under graph structures and temporal encoding to form stable constraints in short and medium term ranges. This reduces the influence of local noise on prediction. As the horizon extends, constraints imposed by inter-node correlations on future states gradually weaken. Uncertainty from cross-node fluctuation propagation accumulates. Errors, therefore, become more likely to increase. This observation suggests that cloud traffic forecasting requires a balance between structural modeling and long-range temporal modeling to maintain robustness in long-horizon prediction.

For cloud resource management applications, horizon sensitivity directly determines the decision time scales that predictions can support. Short horizons are more suitable for high-frequency control scenarios such as dynamic bandwidth allocation, congestion warning, and rapid elastic scaling. Longer horizons are more aligned with medium and long-term strategies such as capacity planning and resource reservation. However, they impose higher requirements on long-term generalization and drift adaptation. Therefore, this analysis guides the selection of appropriate prediction horizons and scheduling frequencies in cloud platforms. It also reinforces the significance of the proposed spatiotemporal structured representation learning approach for improving the reliability of cloud traffic fluctuation prediction.

The learning rate, which controls the magnitude of parameter updates, is a key factor affecting the convergence behavior and stability of time series prediction models. By adjusting the learning rate while

keeping other configurations unchanged, we can examine the model's ability to adapt to changes in the optimization step size and the resulting response to errors. The experimental results are shown in Figure 3.

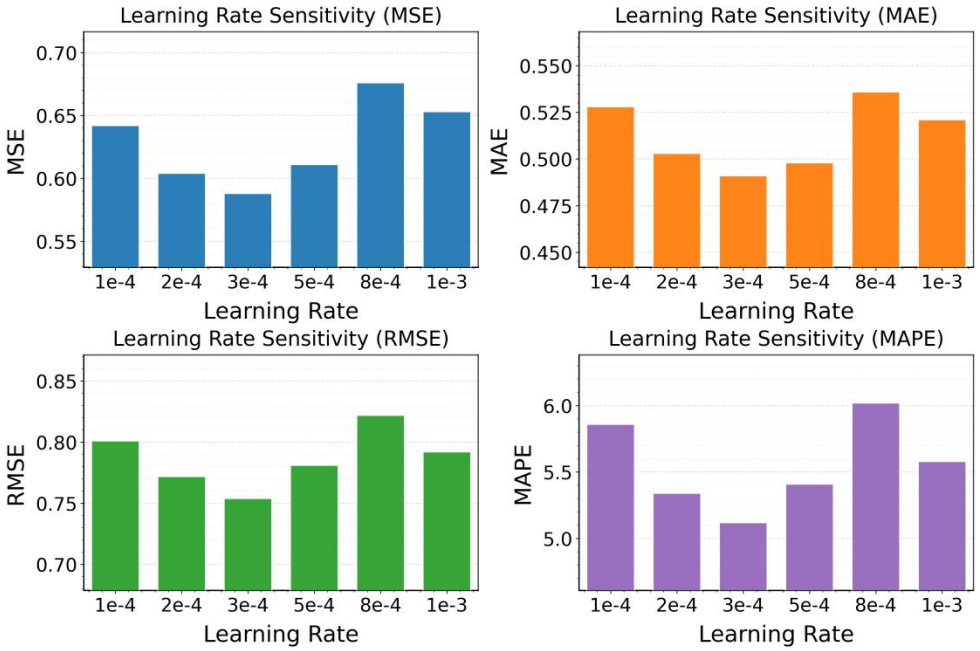

Figure 3. The impact of the learning rate on experimental results

The overall patterns induced by learning rate variation indicate a strong dependence of cloud traffic fluctuation prediction on the optimization step size. Updates that are too small or too large both weaken the model's ability to fit non-stationary sequences. This sensitivity reflects the training difficulty caused by burstiness and noise perturbations in cloud traffic. The model must strike a balance between stable convergence and rapid adaptation to learn generalizable evolution patterns from spatiotemporally coupled signals over complex topologies.

From the curves and bar distributions of error versus learning rate, medium-range learning rates are more likely to yield stable optimization trajectories. This allows the model to establish more consistent mappings between short-term fluctuations and medium-term trends. In contrast, when the learning rate deviates from this stable range, training is more prone to underfitting or oscillatory updates. This degrades the ability to track peak segments and amplifies prediction errors during periods of intense traffic fluctuation. Such behavior is consistent with practical cloud scenarios where hotspot migration and scheduling disturbances occur frequently.

The trends observed under different error metrics are not fully aligned. This indicates that the learning rate influences not only overall fitting quality but also the trade-off between local amplitude errors and relative proportional errors. For cloud traffic prediction, this implies that learning rate selection affects resource management risk preferences. Examples include conservative reservation for burst peaks and tendencies toward over-provisioning during low load periods. Therefore, sensitivity analysis during training is necessary to identify a stable operating range. This helps reduce policy oscillation and misjudgment risks after deployment.

Overall, these results highlight the decisive role of optimization hyperparameters in releasing model representation capacity within a graph-based spatiotemporal modeling framework. Graph convolution aggregation amplifies cross-node correlations, while temporal encoding accumulates long-range dependencies. Improper learning rate settings can therefore cause errors to be magnified through multi-layer propagation. By identifying a more robust learning rate interval, model adaptability to topological dynamics

and workload drift can be enhanced. This provides more reliable predictive support for bandwidth scheduling, congestion warning, and elastic scaling in cloud networks.

In practical deployments, cloud computing traffic fluctuation prediction often faces sets of monitoring objects and communication entities of varying sizes. Changes in the number of nodes directly alter the density of the graph structure and the range of information propagation. By adjusting the node size while keeping other configurations unchanged, the model's stability in modeling structural dependencies under different graph sizes can be tested. This setting also helps evaluate the response of prediction errors to changes in system size in cloud environments with frequent topology expansion and contraction, and service additions and removals. The experimental results are shown in Figure 4.

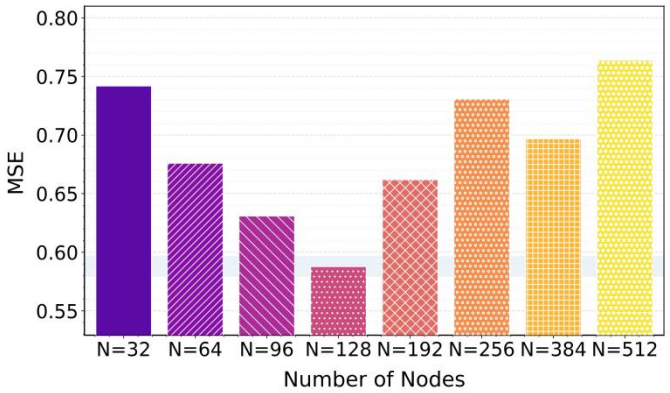

Figure 4. Experiment on the sensitivity of changes in the number of nodes to MSE

When the number of nodes is small, the prediction error is relatively higher. This indicates that an overly small graph scale weakens the coverage of structural context in cloud traffic fluctuation prediction. The model cannot fully capture cross-node dependency propagation and shared fluctuation patterns. Traffic in cloud environments is jointly driven by service invocation chains and shared resource contention. With too few nodes, the graph structure fails to adequately represent real interaction relationships. As a result, error accumulation is more likely to occur during traffic bursts or hotspot migration.

As the number of nodes increases to a moderate scale, the error decreases significantly and reaches a more favorable range. This suggests that, at this scale, the model can better exploit the neighborhood aggregation mechanism of graph neural networks. Local communication relationships and global fluctuation trends are more effectively integrated. This observation aligns with the characteristics of cloud platforms where multiple tenants and services run in parallel. Sufficient node coverage is required to reflect inter-service dependencies and competition. Predictions, therefore, rely not only on individual historical sequences but also on structural information to constrain future evolution. This improves the ability to characterize non-stationary fluctuations.

When the node count continues to grow to a larger scale, the error increases again and exhibits stronger variability. This reflects the simultaneous increase of structural complexity and noise interference in large-scale graph settings. A larger number of nodes is often associated with sparser topologies or more imbalanced edge distributions. Information propagation paths become longer. Local aggregation may introduce background disturbances that are weakly related to target nodes. This makes it more difficult for the model to stably learn key dependencies. In addition, cloud topologies change frequently. Increasing scale amplifies uncertainty caused by structural drift. Prediction errors, therefore, become more sensitive to sampling bias and relationship reconstruction.

Overall, this sensitivity analysis indicates that node scale is a key factor affecting the performance of graph neural network-based traffic prediction. There exists an effective scale range that is more suitable for modeling. In the context of this study, the results further support the rationality of modeling cloud traffic as

graph signals. They also suggest that practical deployment should consider monitoring granularity and topology construction strategies for scale control. Approaches such as node aggregation or selection of key communication entities can be used to maintain effective structural density. This helps preserve graph expressiveness while suppressing noise and instability caused by scale expansion. As a result, the robustness and usability of cloud traffic fluctuation prediction can be improved.

6. Conclusion

This study focuses on key characteristics of cloud environments, including strong traffic fluctuations, complex structural coupling, and temporal nonstationarity. A graph-oriented traffic prediction modeling approach is proposed to characterize traffic evolution from the perspective of overall system operation. By abstracting the cloud network as a time-varying graph and integrating inter-node dependencies with temporal dynamics within a unified framework, the model can more effectively exploit cross-node interaction information. This alleviates the limitation of traditional sequence-based methods in utilizing structural information under complex cloud scenarios. Comparative results demonstrate that the proposed approach achieves more stable prediction performance across multiple error metrics. This confirms the effectiveness of structure-aware spatiotemporal modeling for cloud traffic prediction.

From a methodological perspective, the study shows that introducing graph structures into traffic prediction not only improves overall accuracy but also enhances model adaptability to bursty fluctuations and workload migration. Cloud traffic is often driven by concurrent services, resource contention, and scheduling policy changes. Its fluctuations exhibit clear propagation and coupling effects. Graph neural network-based modeling explicitly represents these propagation relationships at the representation level. The model, therefore, maintains better stability when facing complex topologies and dynamic operating states. This provides an effective modeling approach for latent associations that are difficult to observe directly in cloud networks.

From an application perspective, the findings have direct practical significance for cloud resource management and intelligent operations. More reliable traffic predictions provide forward-looking decision support for bandwidth allocation, congestion avoidance, and elastic scaling. This reduces the risks of resource over-provisioning and service performance degradation. In addition, systematic modeling of traffic evolution helps operations systems identify potential risk regions in advance. This improves the resilience of cloud platforms under high load and burst scenarios. A prediction-centered proactive management paradigm can thus promote the transition of cloud infrastructure from reactive response to intelligent control.

Looking ahead, graph-structured traffic prediction frameworks offer substantial opportunities for further extension in cloud computing and related fields. With the continued development of cloud native architectures and edge computing, coordination across data centers and multi-layer networks will further diversify traffic patterns. Structure-aware modeling is expected to play an important role at larger scales and in more complex environments. Moreover, deep integration of such prediction models with automated scheduling, resource orchestration, and service quality assurance mechanisms may enable adaptive and self-optimizing cloud management systems. This can provide more stable and efficient operational support for large-scale distributed applications, artificial intelligence services, and emerging digital infrastructure.

References

[1] Chen H, Rossi R A, Mahadik K, et al. Graph deep factors for forecasting with applications to cloud resource allocation[C]//Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. 2021: 106-116.

[2] Li J, Yang D, Guo H, et al. Forecasting resource usage pattern changes in clouds via contrast graph-evolution learning[J]. Future Generation Computer Systems, 2024, 154: 373-383.

[3] B. Yu, H. Yin, and Z. Zhu, "Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting," in Proc. IJCAI, 2018, pp. 3634-3640.

[4] Z. Wu, S. Pan, G. Long, J. Jiang, C. Zhang, and Y. Yu, "Graph WaveNet for Deep Spatial-Temporal Graph Modeling," in Proc. IJCAI, 2019, pp. 1907-1913.

[5] H. Wu, J. Xu, J. Wang, et al., "Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting," in Proc. NeurIPS, 2021.

[6] H. Shao, H. So, F. D. Salim, et al., "Informer+: A Unified Framework for Time-Series Forecasting," arXiv:2308.07820, 2023.

[7] Feng B, Ding Z. Application-oriented cloud workload prediction: A survey and new perspectives[J]. Tsinghua Science and Technology, 2024, 30(1): 34-54.

[8] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.

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

[11] Liu Y, Hu T, Zhang H, et al. itransformer: Inverted transformers are effective for time series forecasting[J]. arXiv preprint arXiv:2310.06625, 2023.

[12] H. Wu, T. Xu, Y. Wang, et al., "TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis," in Proc. ICLR, 2023.