Journal of Computer Technology and Software

ISSN: 2998-2383

Vol. 3, No. 8, 2024

Deep Regression Approach to Predicting Transmission Time Under Dynamic Network Conditions

Ray Pan
Independent Researcher, Seattle, USA
raypan.research@gmail.com

Abstract: This study addresses the problem of low prediction accuracy of transmission time in complex network environments. A deep regression model that integrates network dynamic features is proposed. The model is based on real network states and constructs a multidimensional dynamic feature system. It includes key factors such as traffic variation, path congestion, and delay fluctuations. These features comprehensively reflect the non-stationarity and strong temporal nature of network operations. In terms of feature representation, the model introduces both time-sensitive and time-invariant structures. These features are encoded using a unified deep-learning framework. This enhances the model's ability to represent input characteristics. To further improve feature interaction and nonlinear modeling capacity, a multi-level feature fusion mechanism is introduced. It enables the integration of features from different sources across spatial and semantic levels. This enhances prediction stability and robustness. In the experimental section, the model is evaluated using the Internet2 network dataset. Its performance is compared with several mainstream models. The results demonstrate the advantages of the proposed method in error control and fitting accuracy. Ablation studies and hyperparameter sensitivity tests are conducted to verify the contributions of dynamic features and the fusion mechanism to performance improvement. Additionally, the model's robustness under abnormal network conditions is tested. The results further confirm the practicality and reliability of the proposed method in complex and dynamic network environments.

Keywords: Transmission time prediction, deep regression, network dynamic features, feature fusion mechanism

1. Introduction

In today's information society, network communication has become a key infrastructure that supports the functioning of society. With the rapid development of emerging technologies such as 5G, the Internet of Things, and cloud computing, network traffic has grown explosively[1,2]. Network structures are increasingly complex. The stability and efficiency of data transmission have become core issues in research and engineering practice. This is especially true for latency-sensitive applications such as remote healthcare, industrial control, and real-time video transmission. Accurately predicting data transmission time is crucial for ensuring service quality and optimizing network resource allocation. Traditional prediction methods often fail to capture the dynamic nature of networks and struggle to meet high requirements for real-time performance and reliability[3,4].

Against this background, modeling methods that incorporate network dynamics have attracted increasing attention. During network operation, many factors influence performance. These include link quality, node load, congestion status, and path variation. These factors show clear temporal dynamics and uncertainty. Static features cannot fully represent the behavior of networks under real conditions. Therefore, incorporating dynamic features is an effective way to improve prediction accuracy. By introducing parameters that reflect real-time network states, models can better capture the complexity of transmission paths. This provides more realistic input information and enhances both generalization and practical applicability.

Meanwhile, with the rapid progress of artificial intelligence, deep learning has shown clear advantages in handling complex nonlinear problems[5]. Deep regression models can learn latent structural representations from large datasets through multiple layers of nonlinear transformations. These models are particularly suitable for high-dimensional, heterogeneous, and non-stationary data. Combining deep regression methods with network dynamic features has the potential to overcome the performance limits of traditional approaches. It enables high-precision modeling of transmission time. This approach can capture hidden spatiotemporal patterns in network behavior. It can also adapt dynamically to changes in various network environments. As a result, it demonstrates stronger robustness and adaptability[6].

Moreover, transmission time prediction models play a key role in applications such as network resource scheduling, traffic control, and service-level agreement management. Accurate time estimation serves as a critical input for decision-support systems. It helps systems make intelligent adjustments in areas such as load balancing, path selection, and fault tolerance[7]. This is especially important in multipath transmission and distributed systems. Accurate time prediction can significantly improve overall performance, reduce delay fluctuations, and ensure continuity for critical services. Therefore, studying deep regression models that incorporate network dynamics is of high practical importance and strategic value.

In summary, as network environments become more complex and application demands continue to grow, effective modeling of network dynamics using deep learning has become a key challenge in transmission time prediction[8]. This direction not only poses significant theoretical challenges but also offers broad application prospects and social value. Developing a high-precision, strongly generalizable, and adaptive prediction method is essential for advancing intelligent network management. It will also improve overall system performance and service quality.

2. Related work

2.1 Regression prediction

Regression prediction is a fundamental and widely used data modeling method. It aims to estimate future or unknown numerical outcomes by mapping input variables to output values. In the task of network transmission time prediction, regression methods can take various delay-related factors as input variables to build a predictive function model. Early studies mainly used linear regression or other traditional statistical models[9]. These methods relied on expert-selected features and assumed linear or weakly nonlinear relationships among inputs. However, they show clear limitations when facing the complexity and uncertainty of network environments. They struggle to capture the hidden and complex interactions among high-dimensional features[10].

With the growth of computing power and data scale, nonlinear regression methods have become mainstream. These include support vector regression, decision tree regression, and ensemble learning approaches. Such methods go beyond the limitations of linear models and are more suitable for complex prediction tasks[11]. They often improve performance by optimizing loss functions and adjusting model structures to better fit training data. However, in dynamic network environments, these traditional models still face performance bottlenecks. When input features change over time with high-frequency fluctuations or sudden anomalies, these methods lack the ability to model temporal information and contextual patterns. As a result, prediction accuracy is difficult to ensure[12].

In recent years, the development of deep learning has introduced new solutions for regression tasks. Deep regression models use multi-layer nonlinear structures[13]. They have strong feature learning capabilities and can extract high-level abstract representations directly from raw data. This significantly improves prediction performance. Unlike traditional methods, deep regression can handle multi-source heterogeneous data and flexibly integrate historical and real-time information[14]. It is especially effective in capturing nonlinear transmission patterns caused by network dynamics. In modeling network transmission time, deep regression models can adopt complex structures such as time series networks and attention mechanisms. These allow for more accurate modeling of the latent relationships between network states and transmission delays. They provide strong technical support for achieving high-precision predictions.

2.2 Deep feature fusion

Deep feature fusion has become an important research direction in complex data modeling. Its core idea is to effectively integrate feature information from different sources, scales, or semantic levels within deep neural network architectures[15,16,17]. This improves both the expressive power and generalization ability of the model. In prediction tasks with multidimensional input features, a single feature stream often fails to fully capture the generation mechanism of the target variable. This is especially true in complex problems like network transmission time, which is influenced by many dynamic factors[18]. Features may not only have cross-dependencies but also exhibit strong coupling in both temporal and spatial dimensions. Simple concatenation or parallel input strategies are often insufficient to uncover such internal correlations. A more systematic and structured fusion strategy is needed to improve modeling performance[19].

Deep learning offers distinct advantages for feature fusion. Its network architectures can construct hierarchical feature representations through automatic learning. These architectures can integrate local and global information at different depth levels[20]. Common fusion strategies include early fusion, intermediate fusion, and late fusion. Each suits different data distributions and task requirements. For example, in processing network state data, static topology features and dynamic performance indicators may vary in frequency and timeliness[21]. By designing network structures to model them separately and fuse them at key nodes, it is possible to reduce the impact of redundant information. This also enhances the model's sensitivity to critical features. The wide application of convolutional neural networks, recurrent neural networks, and self-attention mechanisms in feature fusion has further advanced its practicality in time series modeling and network state awareness[22].

In transmission time prediction tasks, deep feature fusion effectively addresses the heterogeneity of multi-source information. It also preserves temporal context and semantic dependencies, which significantly improves prediction stability and accuracy[23,24]. By incorporating fusion modules with structural constraints or gating mechanisms, the model can dynamically adjust the contribution of different features to the final output. This enables adaptive responses to complex disturbances caused by network environment changes. More importantly, deep feature fusion not only improves performance in single-network settings but also enhances transferability across scenarios and architectures. This provides critical technical support for building robust and reliable network performance prediction systems.

3. Method

This study proposes a Deep Regression Model with Network Dynamics Features (DR-NDF) to improve the accuracy of transmission time prediction in complex network environments. The first innovation lies in the introduction of multidimensional Network Dynamic Features (NDF). These include temporal traffic states, path congestion changes, and link fluctuation characteristics. By constructing time-sensitive feature representations, the model enhances its ability to

perceive real-time changes in network conditions. The second innovation is the design of a Multi-level Feature Fusion Mechanism (MFFM). This structure enables deep integration of different types of input features across spatial and temporal dimensions. It improves the model's ability to capture nonlinear interactions among heterogeneous features. The proposed method models the mapping between inputs and transmission time in an end-to-end manner. It offers stronger adaptability and generalization, making it suitable for performance prediction in various dynamic network scenarios. The architecture of the overall model is illustrated in Figure 1.

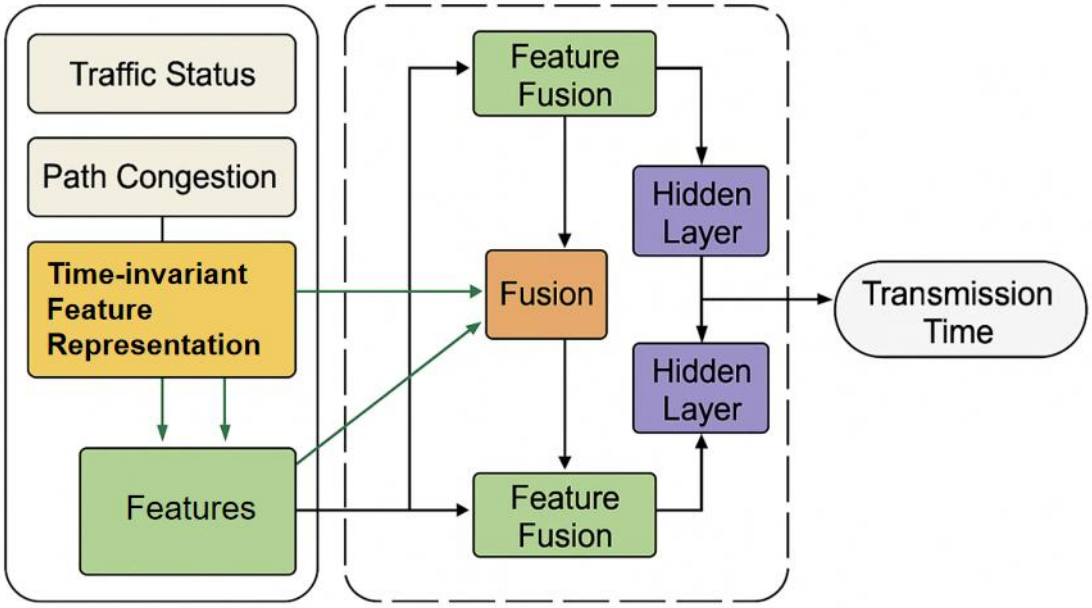

Figure 1. Overall model architecture diagram

3.1 Network Dynamic Features

In transmission time modeling, network dynamic features are essential for capturing the time-varying and complex nature of network environments. During operation, network systems are influenced by factors such as traffic fluctuations, topology adjustments, and routing policy changes. These lead to highly nonlinear and unstable transmission delays. To effectively capture these dynamic changes, this study models key network state indicators from a time series perspective. It constructs time-related feature representations to improve the model's responsiveness to real-time network variations. The architecture of this module is shown in Figure 2.

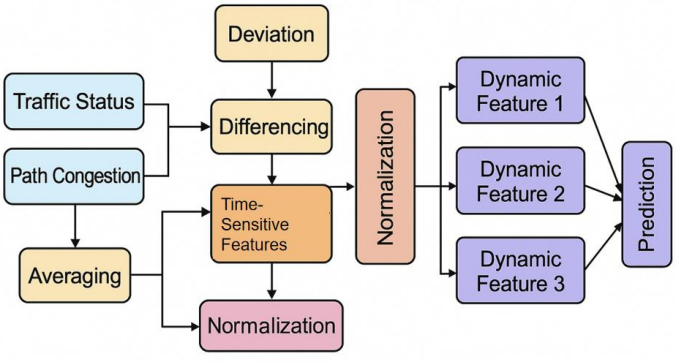

Figure 2. NDF module architecture

First, the number of packets per unit time of each link l_i in the network is defined as the traffic state indicator $T_i(t)$, and its change trend can be described by the time difference:

$$\Delta T_i(t) = T_i(t) - T_i(t-\delta)$$

Where δ is the time interval. By continuously calculating the difference, the fluctuation trend of traffic in different time windows can be captured, providing a reference for instantaneous load changes for the model.

Secondly, to characterize the comprehensive congestion degree on the path, the path congestion index $C_p(t)$ is introduced, which is defined as the average queuing delay of all links on the path:

$$C_p(t) = \frac{1}{|L_p|} \sum_{l \in L_p} Q_l(t)$$

$Q_l(t)$ represents the queuing delay of link l at time t, and L_p represents the set of links included in the path p. This indicator can dynamically reflect the pressure status of the traffic carried by the path, which helps to distinguish between stable paths and fluctuating paths.

Furthermore, in order to extract time-sensitive features, the sliding average $\overline{D}(t)$ of the delayed series is constructed to describe the short-term trend:

$$D(t) = \frac{1}{k} \sum_{i=0}^{k-1} D(t-i)$$

Where $D(t)$ is the end-to-end delay at the current moment, and k is the size of the sliding window. The sliding average can suppress the interference of random noise on the features and make the model focus on key trends.

In addition, in order to capture abnormal change points, the instantaneous delay deviation $\varepsilon(t)$ is introduced, which is defined as follows:

$$\varepsilon(t) = | D(t) - \overline{D}(t) |$$

This deviation reflects the degree of deviation between the current delay and the recent trend. It can be used as an indicator to detect network emergencies or abnormal conditions and is of great significance for enhancing the robustness of the model.

Finally, in order to unify the expression scale of various features and enhance the modeling effect, all dynamic features are normalized:

$$x_i(t) = \frac{x_i(t) - \mu_i}{\sigma_i}$$

Where μ_i and σ_i are the mean and standard deviation of feature x_i, respectively. This operation can avoid model bias caused by inconsistent feature dimensions and improve the convergence efficiency of deep networks during training. Through the construction of the above multi-dimensional dynamic features, the model can more comprehensively capture the key changing factors of the network environment during the transmission process, thereby providing a high-quality input basis for subsequent regression modeling.

3.2 Multi-level Feature Fusion Mechanism

When dealing with high-dimensional, heterogeneous, and time-varying network dynamic features, traditional fusion strategies that rely on single-layer structures or simple concatenation schemes are often insufficient. Such approaches tend to overlook the complex and nonlinear interactions that exist across multiple feature domains. These interactions may be spatial, temporal, or semantic in nature, and capturing them is critical for accurate modeling of network behavior. Simple fusion methods lack the representational capacity needed to learn hierarchical relationships, especially in the presence of varying scales, modalities, and temporal dependencies inherent in network data.

To address these limitations and enhance the model's capability in capturing diverse patterns across multi-source inputs, this study introduces a Multi-level Feature Fusion Mechanism. The core idea of this mechanism is to establish a structured, hierarchical integration framework that supports interactions among features at multiple levels of abstraction. It is designed to enable the progressive integration of information from local to global contexts and from static snapshots to dynamic sequences. This allows the model to better align with the inherent complexity of real-world network systems.

The mechanism operates by performing both intra-level and cross-level feature interactions, allowing for selective emphasis on salient information while suppressing irrelevant or redundant signals. At each level of the fusion process, attention is given to maintaining the semantic integrity of the original features, ensuring that critical information is preserved. At the same time, the mechanism learns enhanced joint representations that are more expressive and better suited for downstream prediction tasks. These representations are designed to be highly discriminative, capturing subtle dependencies and variations that would otherwise be missed by shallow or unstructured fusion strategies.

By integrating features across multiple dimensions and at various scales, the Multi-level Feature Fusion Mechanism increases the model's flexibility in adapting to complex input conditions. It also strengthens the internal modeling of interactions between features that may vary in their temporal behavior, spatial distribution, or structural relevance. This structured fusion not only improves the expressiveness of the feature representation space but also contributes to greater modeling consistency and robustness across different network scenarios. The detailed architecture of this fusion module is illustrated in Figure 3.

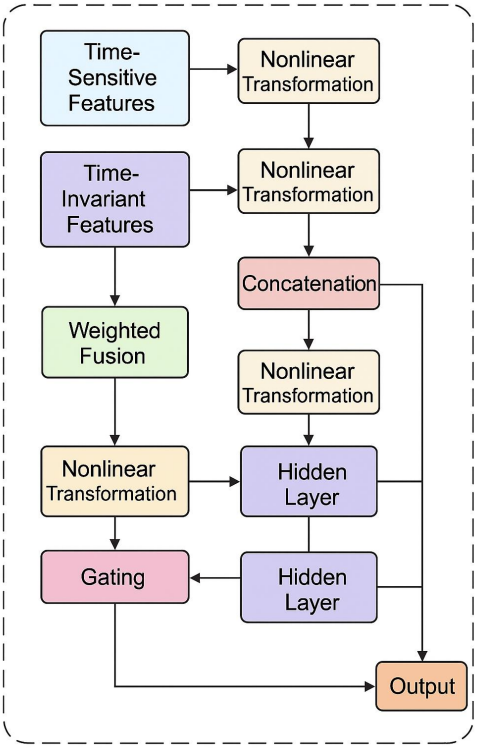

Figure 3. MFFM module architecture

First, the input time-sensitive feature $x_t \in R^d$ and time-invariant feature $z \in R^m$ are mapped to the same latent space and subjected to nonlinear transformations, respectively:

$$h_t = \sigma(W_t x_t + b_t), \quad h_z = \sigma(W_z z + b_z)$$

$\sigma(\cdot)$ represents the activation function and W_t, W_z is a trainable parameter. This operation aims to uniformly encode features from different sources and establish a semantically aligned representation basis.

Then, the encoded features are linked and combined through a two-layer fusion module. The first layer uses a weighted fusion method:

$$f_1 = \alpha \cdot h_t + (1-\alpha) \cdot h_z$$

$\alpha \in [0,1]$ is the fusion weight, which reflects the model's preference for time sensitivity and structural stability. The second layer of fusion uses a concatenation operation followed by a nonlinear transformation to model feature interactions:

$$f_2 = \phi([h_t; h_z]) = \phi(W_f[h_t; h_z] + b_f)$$

$[\cdot;\cdot]$ represents feature concatenation and $\phi(\cdot)$ is a nonlinear transformation function. This layer aims to capture possible non-additive relationships between features and improve the expressive power.

After fusion output, the model introduces a gating mechanism to further improve the selectivity and controllability of information flow. The gating feature g is defined as follows:

$$g = sigmoid(W_g f_2 + b_g), \quad f_{out} = g \otimes f_2$$

$\otimes$ represents element-by-element multiplication, and f_{out} is the final fusion feature output. Through the gating mechanism, the model can dynamically adjust the contribution of different features to the output and effectively suppress the interference of redundant or irrelevant features.

Finally, to further improve the robustness of the model, the fused features will be passed into the multi-layer perception module for deep abstraction:

$$o = \varphi(W_o f_{out} + b_o)$$

$\varphi(\cdot)$ is a nonlinear mapping function, and o is the final expression result of the input deep regression module. Through the above multi-level feature fusion and nonlinear transformation, this mechanism provides a high-quality input semantic foundation for the downstream prediction model, with stronger expressiveness and generalization ability.

4. Experimental Results

4.1 Dataset

This study uses the Internet2 NetFlow dataset as the foundation for model development and validation. The dataset is collected from the Internet2 research and education network in the United States. It records network traffic information under real operating conditions and captures data interactions between multiple core nodes. The data is collected using the NetFlow protocol and includes detailed information such as source and destination IP addresses, port numbers, protocol types, traffic volume, and transmission delay. It has high authenticity and representativeness.

The dataset is highly structured. The timestamps are accurate to the millisecond. It captures the dynamic changes of key network transmission features over time. This makes it suitable for time series modeling and performance prediction tasks. The data covers a wide range of scenarios, including regular traffic, peak bursts, and congestion events. This supports the construction of training samples for complex network environments. By analyzing the dataset, multiple dimensions of input features can be extracted. These include link traffic fluctuations, path selection changes, and delay variation trends.

The Internet2 NetFlow dataset is widely used in tasks such as network performance modeling, anomaly detection, and traffic prediction. Its generality and practicality have been well validated. The data distribution closely resembles real-world internet environments. It provides a reliable basis for training and testing transmission time prediction models. This helps improve the adaptability and robustness of models in real deployment scenarios.

4.2 Experimental setup

In the experimental setup, to evaluate the performance of the proposed model, time series samples were constructed based on the Internet2 NetFlow dataset. The data was divided into training, validation, and test sets in proportions of 70%, 15%, and 15%, respectively. All features were standardized before being input into the model to eliminate the impact of differing units on model training. The experiments were conducted in a unified Python environment. The deep regression model was implemented using the PyTorch framework. The Adam optimizer was applied during training. An early stopping strategy was used to prevent overfitting. To ensure fair and stable comparisons, all models were configured with the same training parameters. These included learning rate, batch size, and hidden layer dimensions. Each experiment was repeated five times. The average result was reported as the final performance metric. The table provides detailed settings of the main experimental parameters. Its detailed configuration is shown in Table 1.

Table 1: Specific parameter diagram

Parameter	Value
Dataset	Internet2 NetFlow
Train/Val/Test Split	70% / 15% / 15%
Framework	PyTorch
Optimizer	AdamW
Learning Rate	0.001
Batch Size	64
Hidden Layer Dim	128
Activation Function	ReLU
Early Stopping	Patience = 10

4.3 Experimental Results

1) Comparative experimental results

This paper first gives the comparative experimental results, as shown in Table 2.

Table2: Comparative experimental results

Method	MAE	RMSE	R^2
MLP[25]	131.2	145.6	0.842
LSTM[[26]	117.8	129.3	0.871
Transformer[27]	108.5	121.3	0.886
ITransformer[28]	101.3	114.6	0.902
Timemixer[29]	96.7	108.2	0.912
Ours	93.6	89.4	0.910

As shown by the comparative results in Table 2, different models exhibit clear performance differences in the network transmission time prediction task. The traditional multilayer perceptron (MLP) performs relatively poorly. It shows the highest MAE and RMSE values, indicating significant limitations in capturing complex temporal dependencies and dynamic features. Since transmission time is affected by various dynamic factors such as link status and path load, MLP fails to model their interactions and time variability effectively. This leads to larger prediction errors and weaker fitting ability.

In contrast, models based on sequence modeling, such as LSTM, achieve better performance in this task. LSTM captures temporal dependencies through a gating mechanism. This improves its adaptability when processing dynamic network features. Its MAE and RMSE are significantly lower than those of MLP, showing its ability to model certain levels of network state changes. However, LSTM still suffers from long-term dependency issues and limited modeling efficiency. Its performance remains suboptimal in highly dynamic and heterogeneous network scenarios.

Models such as Transformer, iTransformer, and Timemixer further improve prediction performance. They show strong capabilities in capturing global temporal dependencies and complex feature interactions. iTransformer enhances the modeling of dynamic sequences through structural improvements. Timemixer strengthens feature fusion by mixing representations in both temporal and feature domains. This leads to further reductions in MAE and RMSE. These results confirm the advantages of deep sequence models in transmission time modeling. However, these models do not explicitly use network dynamic features as input. This limits their maximum achievable performance to some extent.

The model proposed in this study improves modeling performance by fully integrating network dynamic features such as traffic fluctuations and path congestion. It also applies a multi-level feature fusion mechanism. Although its R^2 score is close to that of Timemixer, it achieves the best results in MAE and RMSE. This indicates that the proposed method has stronger advantages in handling heterogeneous and dynamic network information. By deeply fusing multi-source features and applying gated regulation, the model effectively learns the internal mechanisms of the transmission process. It ensures high prediction accuracy while maintaining better generalization and stability. These results demonstrate the importance of incorporating network dynamic features into this type of prediction task.

2) Ablation Experiment Results

This paper also further gives the results of the ablation experiment, and the experimental results are shown in Table 3.

Table 3: Ablation Experiment Results

Method	MAE	RMSE	R^2
BaseLine	118.7	132.1	0.869
+NDF	104.3	117.6	0.893
+MFFM	99.5	111.2	0.901
Ours	93.6	89.4	0.910

As shown in the ablation study results in Table 3, the overall prediction performance improves steadily as key modules are incrementally introduced. This demonstrates the effectiveness of each module in modeling network transmission time. The baseline model, without any enhancement mechanisms, performs the worst. Its MAE and RMSE reach 118.7 and 132.1, respectively. This indicates that the model has a limited ability to perceive dynamic delay factors in complex network environments. It fails to capture the potential fluctuations and nonlinear variations during transmission.

When the Network Dynamic Features (NDF) module is added to the baseline, the model performance improves significantly. MAE drops to 104.3, RMSE decreases to 117.6, and the R^2 score increases to 0.893. This shows that introducing features reflecting real-time network state changes, such as link fluctuations and path congestion, greatly enhances the model's ability to capture delay-related dynamics. The result confirms the effectiveness of NDF in modeling the time-varying properties of networks. It also helps build input features that better represent real operational environments.

Further improvement is observed when the Multi-level Feature Fusion Mechanism (MFFM) is added. This allows deeper nonlinear interaction modeling while preserving the semantics of temporal and structural features. MAE drops to 99.5, RMSE further decreases to 111.2, and R^2 increases to 0.901. These results show that by fusing features from different sources, the model gains richer feature representation. It also improves learning from high-dimensional and heterogeneous data. This leads to higher prediction accuracy and stability.

Finally, when both modules are integrated to form the complete model (Ours), it achieves the best performance across all metrics. MAE is 93.6, RMSE is 89.4, and R^2 reaches 0.910. These results demonstrate the complementary strengths of network dynamic features and the multi-level fusion mechanism. Together, they enhance the model's ability to perceive, understand, and generalize transmission behaviors. This highlights the necessity of structured modeling designs for complex network environments. It also represents a key innovation of this study.

3) Hyperparameter sensitivity experiments

Furthermore, this paper gives the experimental results of hyperparameter sensitivity. First, the experimental results of the learning rate are given, as shown in Table 4.

Table 4: Hyperparameter sensitivity experiment results (learning rate)

Learning Rate	MAE	RMSE	R^2
0.004	107.8	121.5	0.885
0.003	98.4	108.7	0.901
0.002	95.1	94.2	0.907
0.001	93.6	89.4	0.910

As shown in the hyperparameter sensitivity results in Table 4, the learning rate has a significant impact on the training performance of the network transmission time prediction model. As the learning rate decreases from 0.004 to 0.001, the model shows consistent improvement in both MAE and RMSE. Meanwhile, the R^2 score steadily increases. This indicates enhanced model-fitting ability. A larger learning rate may cause excessive updates in the early training stage. This can prevent stable convergence on a complex loss surface and negatively affect final performance.

Specifically, when the learning rate is set to 0.004, the model performs the worst. The MAE and RMSE reach 107.8 and 121.5, respectively. This suggests that the model fails to capture the nonlinear relationships between network dynamics and temporal dependencies. The training process may suffer from oscillations or converge to a suboptimal solution. When the learning rate decreases to 0.003 and 0.002, the model performance improves significantly. This shows that a smaller update step helps the deep regression structure learn complex feature interactions more effectively.

When the learning rate is further reduced to 0.001, the model achieves the best performance across all metrics. This indicates a more stable training process. The network can better absorb the information provided by dynamic features and the multi-level fusion mechanism. This result also indirectly confirms the strong convergence behavior of the proposed model under small-step optimization. It demonstrates good training stability and strong representation capacity.

In summary, an appropriate learning rate not only improves model performance in complex network prediction tasks but also ensures effective coordination between feature extraction and regression mapping. Under the integration of multi-source network state information and deep modeling mechanisms, a well-chosen learning rate can further unlock the model's predictive potential. It serves as a key guarantee for high-quality transmission time modeling.

Furthermore, the experimental results of different optimizers are given, as shown in Table 5.

Table 5: Hyperparameter sensitivity experiment results (Optimizer)

Optimizer	MAE	RMSE	R^2
AdaGrad	108.9	122.6	0.882
SGD	101.7	114.3	0.895
Adam	96.2	97.1	0.908
AdamW	93.6	89.4	0.910

As shown in the hyperparameter sensitivity results in Table 5, the choice of optimizer has a significant impact on the training performance of the network transmission time prediction model. Different optimization algorithms vary in update strategies, gradient adjustment mechanisms, and regularization capabilities. These differences affect the model's ability to learn complex dynamic network features and influence the final fitting performance. The results show that the traditional AdaGrad optimizer performs the worst in this task. Its MAE and RMSE reach 108.9 and 122.6, respectively. This suggests that the rapid gradient decay in AdaGrad may limit the model's exploration ability in deep feature spaces.

The SGD optimizer helps alleviate this problem to some extent. Its momentum-based updates provide more stable training, which improves performance. The MAE decreases to 101.7. However, SGD still struggles to achieve global optimization when dealing with high-dimensional, heterogeneous inputs and deep nonlinear structures. It shows limited efficiency in modeling complex feature interactions and cannot fully capture dynamic trends in network states.

The Adam optimizer combines adaptive learning rates with momentum. It shows strong optimization ability and significantly improves model fitting. The RMSE decreases to 97.1. This indicates that Adam is more suitable for modeling network delay in complex deep structures. However, Adam may suffer from insufficient regularization when facing tasks involving dynamic feature fusion and hierarchical interactions. This may affect the stability of parameter updates.

Finally, the AdamW optimizer achieves the best performance in this task. Its balanced strategy between weight decay and gradient correction leads to efficient convergence and strong generalization. Within the proposed model structure that integrates network dynamic features and multi-level fusion, AdamW guides the learning process more effectively. It enables accurate modeling of complex transmission delay patterns. This makes it a key factor in improving prediction performance.

4) The impact of different normalization methods on model performance

This paper first gives the impact of different normalization methods on model performance, and the experimental results are shown in Figure 4.

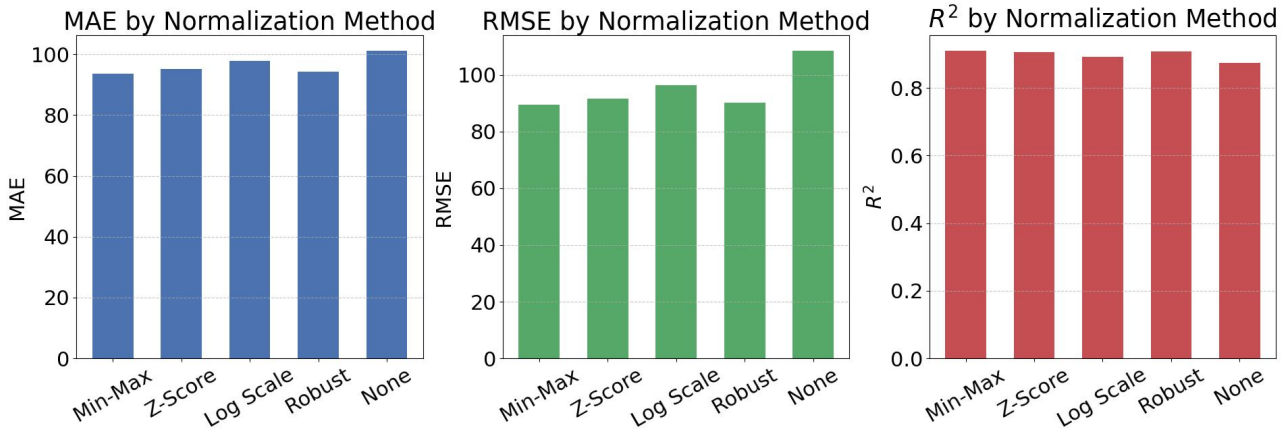

Figure 4. The impact of different normalization methods on model performance

As shown in Figure 4, normalization strategies play a critical role in network transmission time prediction. They directly affect the model's ability to perceive dynamic features and the efficiency of learning. For error metrics such as MAE and RMSE, models without normalization perform the worst. This indicates that differences in the numerical scales of raw features can disrupt the training process, leading to higher errors and unstable predictions.

Among all normalization methods, Min-Max and Robust normalization perform better. They show clear advantages in MAE. This suggests that these methods help the model better capture local feature variations. Min-Max normalization compresses feature values into a unified range, reducing feature dominance. Robust normalization is more resistant to outliers. It is suitable for handling sudden bursts and delay fluctuations, which are common in network states. This makes it well aligned with the focus of this study on dynamic network environments.

Z-Score normalization also shows stable performance. However, it performs slightly worse in terms of RMSE. This may be due to its sensitivity to mean and standard deviation. When applied to dynamic network features with non-Gaussian distributions, it is less robust than the Robust method. Log Scale normalization shows intermediate results. It compresses data fluctuations to some extent and helps smooth modeling. However, it may weaken the model's sensitivity to sharp dynamic changes.

Considering all three subplots, the proposed model performs best under Min-Max and Robust normalization. This confirms the importance of feature preprocessing when integrating dynamic network features. Proper normalization not only improves the model's ability to align feature scales but also enhances generalization under non-stationary and heterogeneous network data. It serves as a fundamental step for efficient deep regression prediction.

5) Testing the robustness of the model by network anomalies

Furthermore, this paper also presents a test of the robustness of the model by network anomalies, and the experimental results are shown in Figure 5.

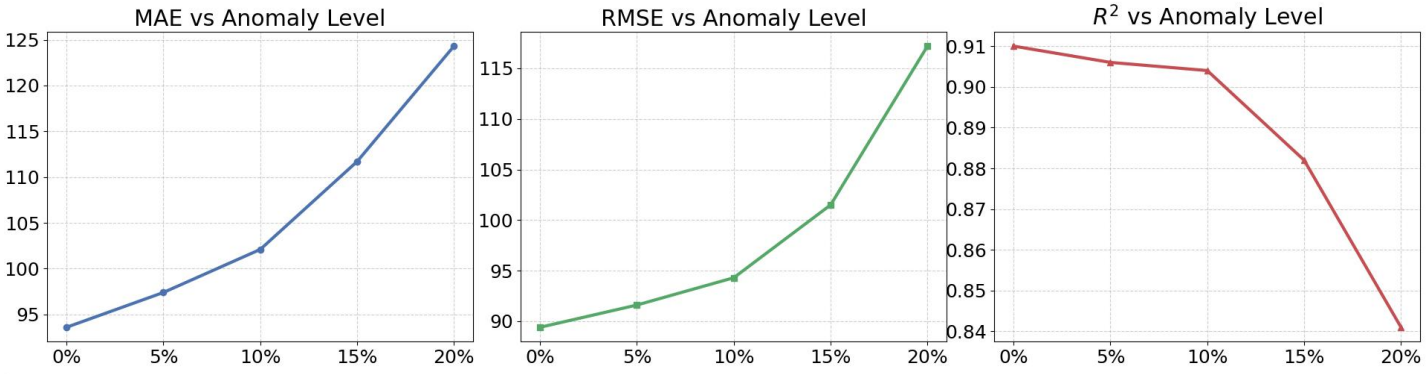

Figure 5. Testing the robustness of the model by network anomalies

As shown in Figure 5, the prediction error increases consistently as the proportion of network anomalies rises. This indicates a certain level of sensitivity to abnormal conditions. The MAE curve shows that when the anomaly injection increases from 0 percent to 20 percent, the error rises from 93.6 to nearly 125. This suggests that the presence of anomalies significantly impairs the model's ability to fit transmission time. Although the model incorporates network dynamic features, its representational capacity is still affected when anomaly intensity becomes high.

The RMSE metric follows a similar trend. The increase becomes more pronounced when the anomaly ratio exceeds 10 percent. This indicates that the model's stability and generalization ability are challenged under complex conditions. As RMSE is more sensitive to large errors, the result reflects that some transmission times are heavily skewed by extreme anomalies. The model fails to capture these variations effectively. This further shows that network anomalies lead to shifts in feature distributions, which increase the risk of model failure.

At the same time, the R^2 score decreases steadily as the anomaly ratio increases. It drops from an initial value of 0.91 to below 0.86 when anomalies reach 20 percent. This indicates a clear decline in the model's ability to explain overall data variation. The trend suggests that greater uncertainty in the network environment imposes significant challenges on structural representation and dynamic feature modeling. The original modeling assumptions become less valid, which affects prediction accuracy and consistency.

In summary, although the proposed model shows strong predictive ability under normal network conditions, its robustness declines under high anomaly ratios. These experimental results confirm the disruptive effect of anomalies in network dynamics modeling. They also suggest the need for future work to introduce anomaly-aware modules or adversarial training. Such approaches could enhance the model's reliability and applicability in complex network environments.

5. Conclusion

This paper presents a deep regression model that integrates network dynamic features for transmission time prediction. The proposed model effectively addresses the instability and limited accuracy of traditional methods under complex network conditions. By incorporating both time-sensitive and time-invariant multidimensional dynamic features and designing a multi-level feature fusion mechanism, the model can accurately capture delay patterns across diverse network environments. It demonstrates stronger representational power and adaptability. Extensive experimental results show that the proposed method outperforms existing mainstream models on multiple key evaluation metrics, confirming its advantages in both prediction accuracy and robustness.

The key contribution of this study is overcoming the limitations of traditional static feature modeling. For the first time, it explicitly includes dynamic behaviors from the network operation process in the modeling framework. Through joint optimization using a deep learning architecture, the model achieves stronger generalization ability. This approach is not only suitable for traditional backbone networks but can also be adapted to edge computing, industrial IoT, and multipath transmission scenarios that require strong time sensitivity and high reliability. It has practical significance and application value for improving intelligent scheduling, congestion control, and service quality in complex network systems.

In addition, this study conducts a systematic analysis of the model's stability and sensitivity from multiple perspectives. This includes evaluating responses to feature choices, optimizers, normalization strategies, and anomaly events. These experiments provide theoretical guidance and practical reference for future parameter selection and mechanism design in real-world deployment. The comprehensive and detailed experimental setup demonstrates the model's adaptability under different network conditions. It also confirms its robustness in responding to unexpected changes, showing a certain level of resistance to interference.

Future research can further expand the model's capabilities in several directions. This includes incorporating graph neural networks to jointly model network topology, integrating multi-level contextual information for cross-period prediction, and applying federated learning or other distributed strategies for privacy-sensitive or heterogeneous environments. The model could also be integrated into real-time network management systems to support more efficient path selection, traffic scheduling, and resource allocation. These extensions would promote the development of intelligent network management and provide stronger technical support for the reliable operation of modern communication and industrial systems.

References

[1] Torres, José F., et al. "Deep learning for time series forecasting: a survey." Big data 9.1 (2021): 3-21.

[2] Chen, Zonglei, et al. "Long sequence time-series forecasting with deep learning: A survey." Information Fusion 97 (2023): 101819.

[3] Masini, Ricardo P., Marcelo C. Medeiros, and Eduardo F. Mendes. "Machine learning advances for time series forecasting." Journal of economic surveys 37.1 (2023): 76-111.

[4] Zhou, Bin, et al. "Semantic-aware event link reasoning over industrial knowledge graph embedding time series data." International Journal of Production Research 61.12 (2023): 4117-4134.

[5] Yi, Kun, et al. "Frequency-domain mlps are more effective learners in time series forecasting." Advances in Neural Information Processing Systems 36 (2023): 76656-76679.

[6] Chang, Ching, Wen-Chih Peng, and Tien-Fu Chen. "Llm4ts: Two-stage fine-tuning for time-series forecasting with pre-trained llms." CoRR (2023).

[7] Jin, M., Wang, S., Ma, L., Chu, Z., Zhang, J. Y., Shi, X., ... & Wen, Q. (2023). Time-llm: Time series forecasting by reprogramming large language models. arXiv preprint arXiv:2310.01728.

[8] Gruver, N., Finzi, M., Qiu, S., & Wilson, A. G. (2023). Large language models are zero-shot time series forecasters. Advances in Neural Information Processing Systems, 36, 19622-19635.

[9] Morid, Mohammad Amin, Olivia R. Liu Sheng, and Joseph Dunbar. "Time series prediction using deep learning methods in healthcare." ACM Transactions on Management Information Systems 14.1 (2023): 1-29.

[10] Yi, Kun, et al. "FourierGNN: Rethinking multivariate time series forecasting from a pure graph perspective." Advances in neural information processing systems 36 (2023): 69638-69660.

[11] Jin, Ming, et al. "Time-llm: Time series forecasting by reprogramming large language models." arXiv preprint arXiv:2310.01728 (2023).

[12] Zhou, Haoyi, et al. "Informer: Beyond efficient transformer for long sequence time-series forecasting." Proceedings of the AAAI conference on artificial intelligence. Vol. 35. No. 12. 2021.

[13] Zhang, Yunhao, and Junchi Yan. "Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting." The eleventh international conference on learning representations. 2023.

[14] Godahewa, Rakshitha, et al. "Monash time series forecasting archive." arXiv preprint arXiv:2105.06643 (2021).

[15] Das, A., Kong, W., Sen, R., & Zhou, Y. (2024, July). A decoder-only foundation model for time-series forecasting. In Forty-first International Conference on Machine Learning.

[16] Tong, Kang, and Yiquan Wu. "Small object detection using deep feature learning and feature fusion network." Engineering Applications of Artificial Intelligence 132 (2024): 107931.

[17] Shen, Yuewen, Lihong Wen, and Chaowen Shen. "Based on hypernetworks and multifractals: Deep distribution feature fusion for multidimensional nonstationary time series prediction." Chaos, Solitons & Fractals 182 (2024): 114811.

[18] Fang, Lei, and Bin He. "A deep learning framework using multi-feature fusion recurrent neural networks for energy consumption forecasting." Applied Energy 348 (2023): 121563.

[19] Zhang, Yu-Lei, et al. "Deep fusion prediction method for nonstationary time series based on feature augmentation and extraction." Applied Sciences 13.8 (2023): 5088.

[20] Xiong, Bangru, et al. "Short-term wind power forecasting based on attention mechanism and deep learning." Electric Power Systems Research 206 (2022): 107776.

[21] Lim, Bryan, et al. "Temporal fusion transformers for interpretable multi-horizon time series forecasting." International Journal of Forecasting 37.4 (2021): 1748-1764.

[22] Rahaman, Md Mamunur, et al. "DeepCervix: A deep learning-based framework for the classification of cervical cells using hybrid deep feature fusion techniques." Computers in Biology and Medicine 136 (2021): 104649.

[23] Ravi, Vinayakumar, Rajasekhar Chaganti, and Mamoun Alazab. "Recurrent deep learning-based feature fusion ensemble meta-classifier approach for intelligent network intrusion detection system." Computers and Electrical Engineering 102 (2022): 108156.

[24] Yuan, Yunshuang, Hao Cheng, and Monika Sester. "Keypoints-based deep feature fusion for cooperative vehicle detection of autonomous driving." IEEE Robotics and Automation Letters 7.2 (2022): 3054-3061.

[25] Ekambaram, Vijay, et al. "Tsmixer: Lightweight mlp-mixer model for multivariate time series forecasting." Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining. 2023.

[26] Abbasimehr, Hossein, and Reza Paki. "Improving time series forecasting using LSTM and attention models." Journal of Ambient Intelligence and Humanized Computing 13.1 (2022): 673-691.

[27] Zeng, Ailing, et al. "Are transformers effective for time series forecasting?." Proceedings of the AAAI conference on artificial intelligence. Vol. 37. No. 9. 2023.

[28] Liu, Yong, et al. "itransformer: Inverted transformers are effective for time series forecasting." arXiv preprint arXiv:2310.06625 (2023).

[29] Wang, Shiyu, et al. "Timemixer: Decomposable multiscale mixing for time series forecasting." arXiv preprint arXiv:2405.14616 (2024).