An Evolving Transformer Network Based on Hybrid Dilated Convolution for Traffic Flow Prediction

Qi Yu[1,2], Weilong Ding[1,2(✉)], Maoxiang Sun[1,2], and Jihai Huang[3]

[1] School of Information Science and Technology, Beijing 100144, China
dingweilong@ncut.edu.cn
[2] Beijing Key Laboratory on Integration and Analysis of Large-Scale Stream Data, Beijing 100144, China
[3] Zhengzhou University of Technology, Zhengzhou 450044, China

Abstract. Decision making based on predictive traffic flow is one of effective solutions to relieve road congestion. Capturing and modeling the dynamic temporal relationships in global data is an important part of the traffic flow prediction problem. Transformer network has been proven to have powerful capabilities in capturing long-range dependencies and interactions in sequences, making it widely used in traffic flow prediction tasks. However, existing transformer-based models still have limitations. On the one hand, they ignore the dynamism and local relevance of traffic flow time series due to static embedding of input data. On the other hand, they do not take into account the inheritance of attention patterns due to the attention scores of each layer's are learned separately. To address these two issues, we propose an evolving transformer network based on hybrid dilated convolution, namely HDCformer. First, a novel sequence embedding layer based on dilated convolution can dynamically learn the local relevance of traffic flow time series. Secondly, we add residual connections between attention modules of adjacent layers to fully capture the evolution trend of attention patterns between layers. Our HDCformer is evaluated on two real-world datasets and the results show that our model outperforms state-of-the-art baselines in terms of MAE, RMSE, and MAPE.

Keywords: Traffic flow prediction · Transformer · Hybrid dilated convolution · Time series · Attention mechanism

1 Introduction

Traffic congestion on highways has always been a major issue in large cities. With the rapid growth of urbanization and car ownership, the volume and complexity of traffic travel data have become increasingly large [14]. Accurate predictive traffic flow guides cost-effective traffic decisions to alleviate road congestion and enhance the efficiency of highway operations [25].

Supported by the Key-Area Research and Development Program of Guangzhou City.

H. Gao et al. (Eds.): CollaborateCom 2023, LNICST 563, pp. 329–343, 2024.
https://doi.org/10.1007/978-3-031-54531-3_18

The traffic flow prediction task refers to extracting useful information from historical data through technical means and then outputting the most likely future traffic flow. Capturing and modeling the dynamic temporal relationships in traffic flow data is an important part of this task. The current traffic flow prediction task has the following characteristics. First, the complexity of the relationships between data increases due to the massive amount of traffic data, so it is difficult to efficiently extract the relevance between key data. Second, in addition to being influenced by historical traffic flow, the traffic flow data on roads is also closely related to the surrounding context of the data points. Third, traffic flow data time series usually exhibit obvious periodicity (hourly, daily, weekly), which is also a problem that researchers need to consider. In recent years, the vanilla transformer networks [26] has demonstrated powerful capabilities in capturing long-range dependencies and interactions in sequences. Compared with traditional deep learning networks such as convolutional neural networks (CNNs) or recurrent neural networks (RNNs), the original transformer (i.e., vanilla transformer) replaces the most commonly used recurrent layer in the encoder-decoder architecture with multi-head self-attention and models sequences entirely based on attention mechanisms, allowing the network to process all input data in parallel to ensure that the model learns the global relevance of time series. In addition, it also proposes a position encoding mechanism to preserve the order of elements in a sequence. Traffic transformer [5] captures the continuity and periodicity of traffic flow time series using a transformer framework. RPConvformer [29] uses one-dimensional convolution to embed traffic flow time series data as a replacement for the word embedding method of the original transformer.

However, existing transformer network-based traffic flow prediction models still have limitations. On the one hand, the static data embedding method cannot enable the model to learn the dynamic relevance of time series well. The increase in convolution operations that can learn the local correlation of sequences is limited by the size of the convolution kernel. On the other hand, most existing attention-based models do not take into account the inheritance of attention patterns between layers. They simply stack encoding modules repeatedly and learn the attention scores of each layer separately.

To address the aforementioned issues, we propose an Evolving Transformer Network based on Hybrid Dilation Convolution, called HDCformer, for traffic flow prediction. First, in order to capture the dynamic relevance of time series while paying attention to the relevance of local areas, we use a hybrid dilated convolution layer to embed the original time series data. Dilated convolution can increase the receptive field on the basis of standard one-dimensional convolution while also solving the problem of reduced accuracy caused by max pooling operations. Second, in order to capture the dependency relationship between attention scores of different layers, we directly connect attention modules from adjacent layers so that attention calculation can depend on the results of the previous layer and promote information sharing between different layers. In summary, the main contributions of this paper are as follows.

- We propose an Evolving Transformer Network based on Hybrid Dilation Convolution. Our approach adopts the vanilla transformer's encoder-decoder architecture, which is based on scaled dot-product attention. We develop a novel convolutional embedding layer that learns the dynamism and local correlations of traffic flow time series data by stacking dilation convolutions with different dilation rates.
- We conduct residual connections to the original transformer structure, connecting the attention scores of adjacent layers to fully capture the transferability of attention values between different layers, thereby better learning the evolution trend of attention patterns.
- We evaluate our model on two public datasets. The experimental results demonstrate that our model outperforms state-of-the-art baselines on three metrics: MAE, RMSE, and MAPE.

The rest of this paper is organized as follows. In Sect. 2, we introduce the related work of this article, including traditional traffic flow prediction methods and traffic flow prediction methods based on transformer networks. In Sect. 3, we introduce some basic symbols and problem definitions. In Sect. 4, we describe the specific implementation details of the traffic flow prediction method proposed in this paper. In Sect. 5, we evaluate the performance of the proposed method on two real datasets through experiments and ablation studies. Finally, we summarize the work of this paper and discuss future research directions.

2 Related Work

2.1 Traffic Flow Prediction

The traffic flow prediction can be seen as a mapping from historical time series traffic data to future time series traffic conditions. Early methods for traffic flow prediction were based on statistical theory, such as the Historical Average (HA) [15], which uses weighted calculations of traffic flow between adjacent time periods and historical periods as the prediction result. The ARIMA [12] captures linear relationships in time series data for traffic flow prediction. Statistical models are simple in algorithm and rely on statistical assumptions, but cannot capture the temporal characteristics and deeper feature information of traffic flow data.

The emergence of machine learning methods has enabled models to begin capturing non-linear relationships in data. The most representative of these is the traffic flow prediction model based on the KNN [8], which selects K nearest vectors from the historical database for statistical analysis to obtain prediction results. The SVR [6] requires the selection of an appropriate kernel function to train the support vector machine and predict traffic flow in the next period. However, machine learning methods are still unable to cope with increasingly large and complex data.

To support the prediction of today's large amounts of traffic flow data, researchers have begun to use deep learning methods to explore and mine deep

temporal features and dynamic dependencies between data [16]. Most traffic flow prediction models are based on CNNs [13] and RNNs [19]. On one hand, due to the architectural characteristics of RNNs, their network depth can be the length of a time series. Many researchers use RNNs to construct dynamic time relationships. LSTM [30] solves the gradient vanishing problem brought by deep RNNs by adding forget gates, input gates, and output gates. SBU-LSTM [10] developed a bidirectional LSTM layer to capture forward and backward dependencies in time series data. LSTM-BILSTM [21] further improves prediction accuracy by combining the advantages of sequential data with the long-term dependencies of bidirectional LSTMs. GRU [9] simplifies the neural network structure by omitting gates with small contributions in LSTM, improving model learning efficiency. However, due to its recursive nature, RNNs models are always limited in solving global parallelization problems in sequences. On the other hand, early on, CNNs were applied to capture spatial dependencies in traffic flow data in grid road networks. Recent work has also applied CNNs to time series prediction. G-CNN [34] explored a feature extractor for high-dimensional multivariate time series. Some researchers have proposed stacking dilated convolutions, or combining them with causal convolutions, such as [3,4,11], they all demonstrate the applicability of convolutional networks to time series prediction problems. Inspired by their works, we developed a hybrid dilated convolution layer to extract dynamic relevance in time series.

2.2 Transformer Networks for Traffic Flow Prediction

The work [2] first used the attention mechanism in the encoder-decoder, applying it to the task of neural machine translation. Subsequently, the attention mechanism was widely used in time series prediction tasks [18,22,24]. The vanilla transformer [26] is a network architecture completely based on self-attention mechanisms. Due to the outstanding performance of transformers in the field of natural language processing, more and more researchers have begun to explore the feasibility of transformers in traffic flow prediction tasks. Traffic transformer [5] captures the continuity and periodicity of time series using the transformer framework. TERMCast [33] proposes a transformer-based urban traffic prediction architecture that extracts proximity, periodicity, and trend components from urban traffic sequences.

In addition, researchers have improved the original transformer from two aspects to achieve optimal prediction performance.The first is to improve several important modules in the transformer framework. For example, LogSparse [17] Transformer proposes a convolutional self-attention mechanism that better incorporates local environments into the attention mechanism. Informer [36] selects major queries based on query and key similarity to participate in attention score calculation, reducing computational complexity. Pyraformer [20] introduces a pyramid attention module to capture a wide range of time dependencies. Crossformer [35] captures cross-dimensional dependencies to achieve multivariate time series prediction.The second is to design a new transformer architecture. For example, an article [28] proposes a convolution-enhanced attention network that

promotes information flow between tokens across layers. Autoformer [31] designs an autocorrelation mechanism that progressively decomposes complex time series to reduce complexity. Scaleformer [23] designs a new iterative scaling scheme that iteratively improves predicted time series at multiple scales. We design convolutional network modules and residual connections to improve the vanilla transformer architecture.

3 Preliminary

Relevant formal definitions are listed here.

- **Definition 1: Traffic Flow Tensor.** We define the traffic flow tensor of all nodes over the total T time slices as $\{X_1, \cdots, X_n, \cdots, X_N\} \in \mathbb{R}^{N \times T}$, where $X_n = \{x_1, \cdots, x_t, \cdots, x_T\} \in \mathbb{R}^T$ represents the historical time series of node n at the last T time steps, N represents the number of nodes in the road network.
- **Definition 2: Traffic Flow Prediction Problem.** The objective of traffic flow forecasting is to accurately predict future traffic flow values utilizing historical data obtained from N nodes. By fitting a complex function $\tilde{f}$, traffic values for the coming P time steps can be forecasted based on traffic data from N nodes over the previous T time steps. The function is defined as shown in Formula 1, where θ represents the learnable parameter that is shared among the time series of all N nodes within the model.

$$[x_{T+1}, \cdots, x_{T+p}, \cdots, x_{T+P}] = \tilde{f}([x_1, \cdots, x_t, \cdots, x_T; \theta]). \tag{1}$$

4 Methodology

Figure 1 illustrates the framework of HDCformer, which consists of a data embedding layer, a positional encoding layer, a stack of M identical encoder-decoder layers, and an output layer. HDCformer is an improved model based on the transformer network, where the modification specifically includes two aspects. On the one hand, hybrid dilated convolution is used to construct the data embedding layer, which captures the correlation of local time series. On the other hand, a residual connection is added between adjacent encoding layers. That enables the encoding layer to generate attention based on inherited information and learn the evolving trend of attention patterns.

4.1 Hybrid Dilated Convolution for Data Embedding

In addition to being influenced by historical ones, traffic flow is also heavily related to the temporal characteristics [17]. The trend brought by these temporal characteristics are reflected in the surrounding context. As shown in Fig. 2(a), points A and B have the same value which is just the median within a one-hour, but they have completely different fluctuation trends in subsequent time steps. In other words, the two points have different local temporal characteristics.

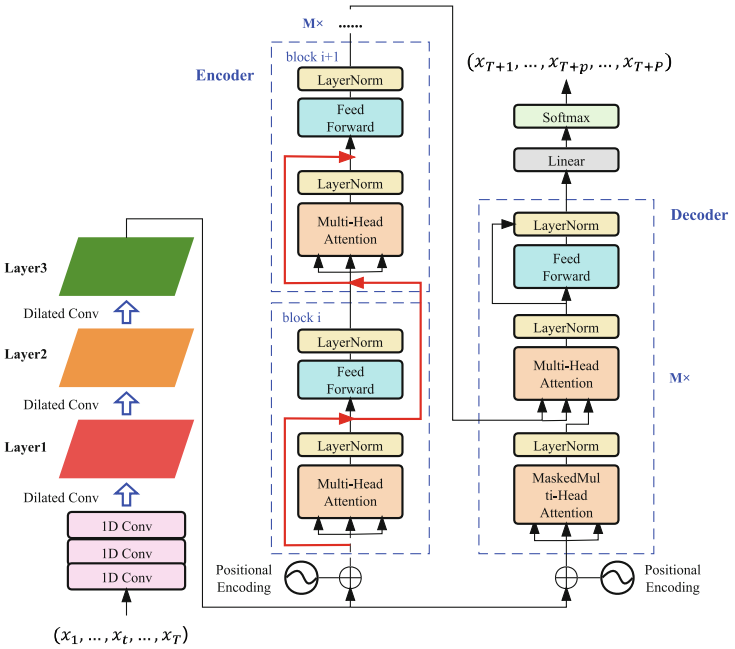

Fig. 1. The framework of HDCformer

For our model, it is necessary to capture such local relevance of time series. In the vanilla transformer network, the attention score is calculated on a point-by-point basis, which may lead to time steps with similar traffic flow data capturing more incorrect matching information. To address this issue, we propose a data embedding layer based on hybrid dilated convolution, which obtains richer local information with different dilation rates.

Dilated convolution, as the name suggests, involves injecting holes into standard convolution to increase the receptive field of the convolutional kernel [32]. The number of holes injected, i.e., the interval between kernels, is referred to as the dilation rate (denoted as r). In standard convolution, the dilation rate r is equal to 1. As can be seen from Fig. 2(b), for a 3×3 convolutional kernel, the setting $r = 2$ significantly increases the receptive field. However, when we stack multiple 3×3 kernels with $r = 2$, we find that our kernel is not continuous. This stacking method results in a loss of information continuity, which contradicts our original intention of learning time series correlations. A feasible solution [27] adopts different dilation rates are set for dilated convolutions at different layers to cover all holes, and achieves better prediction performance than traditional standard convolution.

The implementation of the data embedding layer is shown in Fig. 1. We combine hybrid dilated convolution with one-dimensional standard convolution. The one-dimensional standard convolution unit processes the input time series and

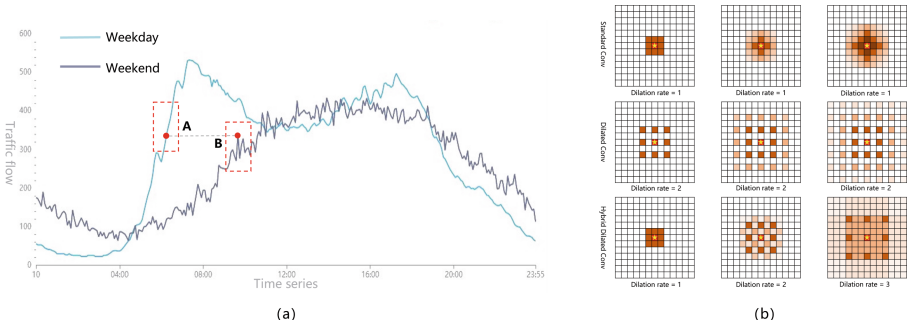

Fig. 2. (a) shows time series on PeMSD4 dataset, where two curves represent the traffic flow of a node on a weekday and a weekend respectively. Each row in (b) shows the pixels involved in the convolutional kernel calculation when the convolutional kernel size is 3×3 for standard convolution (dilation rate = 1), dilated convolution (all dilation rates = 2), and hybrid dilated convolution (dilation rates of 1, 2, and 3), respectively.

sequentially captures local context. By stacking one-dimensional convolution, the sequence features are expanded without altering the sequence length. Then, three layers of hybrid dilated convolution with different dilation rates are used to improve the local correlation of the time series. Assuming that the input traffic flow data is X_{input}, the mathematical expression of the convolution module is shown in Formula 2, where B represents the batch size, T_{in} represents the length of the input sequence, D_{in} represents the input feature dimension, $HDC(\cdot)$ represents the hybrid dilated convolution and D_{emb} represents the expanded feature dimension.

$$X_{input} \in \mathbb{R}^{B \times T_{in} \times D_{in}} \xrightarrow{HDC(\cdot)} X_{emb} \in \mathbb{R}^{B \times T_{in} \times D_{emb}}. \tag{2}$$

Furthermore, to prevent the front position from accessing future information, padding is adopted with a one-sided complement that has the same output sequence.

4.2 Positional Encoding

Due to the parallel nature of the framework, sine and cosine functions were introduced as position encoding in the vanilla transformer framework to mark the order of the input sequence. In text data, position encoding records the position information of words in sentences. Compared to text data, traffic flow data has obvious periodic trend [5]. Therefore, we need to set an appropriate period value in the positional encoding for time series data. The mathematical expression of position encoding is shown in Formula 3, where t represents the time step in the sequence and *period* is the pre-defined parameter.

$$\begin{cases} X_{pos}(2t) & = sin(\frac{2\pi t}{period}) \\ X_{pos}(2t+1) & = cos(\frac{2\pi t}{period}) \end{cases}. \tag{3}$$

We concatenate the above positional encoding vector $X_{pos} \in \mathbb{R}^{B \times T_{in} \times D_{pe}}$ with the data embedding vector $X_{emb} \in \mathbb{R}^{B \times T_{in} \times D_{emb}}$ to obtain the input data $\mathcal{X} \in \mathbb{R}^{B \times T_{in} \times (D_{pe} + D_{emb})}$ for the model, where D_{pe} represents the dimension of positional encoding.

$$\mathcal{X} = Concat(X_{pos}, X_{emb}). \tag{4}$$

The *period* helps to more accurately describe the periodic characteristics of traffic flow data. According to experience, shorter periods can obtain more effective positional encoding values.

4.3 Evolving Transformer

In the vanilla transformer network, the attention scores of each layer are learned separately and no interaction exists between layers. That makes it impossible to learn the evolution trend of attention patterns. Based on the encoder-decoder architecture using Scaled Dot-Product Attention [26] in the traditional transformer, we propose an evolution mechanism that enables stacked blocks to capture the dependency relationship between attention scores at different layers. Accordingly, predictive accuracy can be further improved.

The Evolving Mechanism. Inspired by the work [28], our Evolving Transformer adds residual connections between adjacent encoder blocks. Specifically, the attention score of the previous layer is combined with the output result of the previous block as the input of the current layer's attention mechanism. Then, the calculated attention score of the current layer is combined with the input of the current block and sent to the feedforward network. After layer normalization [1], the output of the current block is obtained. We use $\mathcal{X}_{res}^{i-1}$ to represent the computed result of the previous block after residual connection (before Feed Forward Layer), $\mathcal{X}_{out}^{i-1}$ to represent the output of the previous block, $\mathcal{X}_{in}^{i}$ to represent the input to the attention mechanism of the current block, and $\mathcal{X}_{out}^{i}$ to represent the output of the current block. Further, $Attention(\cdot)$ is used to represent the attention operation on $\mathcal{X}_{in}^{i}$ in the current block. In addition, symbols α and β are pre-defined hyperparameters ranging from 0 to 1 based on empirical values. The mathematical representation is as follows:

$$\begin{aligned} \mathcal{X}_{in}^{i} &= \alpha \cdot \mathcal{X}_{res}^{i-1} + (1-\alpha) \cdot \mathcal{X}_{out}^{i-1}, \\ \mathcal{X}_{res}^{i} &= \beta \cdot \mathcal{X}_{in}^{i} + (1-\beta) \cdot Attention(\mathcal{X}_{in}^{i}), \\ \mathcal{X}_{out}^{i} &= LayerNorm(FeedForward(\mathcal{X}_{res}^{i})). \end{aligned} \tag{5}$$

Multi-head Attention. We use the Scaled Dot-Product Attention model to calculate attention scores, which is a core part of the transformer architecture. First, the input data $\mathcal{X}_{in}^{i}$ is linearly transformed to obtain the initial representations of the three vectors $Q = W_q\mathcal{X}_{in}^{i}$, $K = W_k\mathcal{X}_{in}^{i}$, and $V = W_v\mathcal{X}_{in}^{i}$, which are then input into the model. Next, the dot product is calculated between Q and K to determine their relevance, and then the attention scores are calculated through softmax.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{D_{pe} + D_{emb}}})V. \tag{6}$$

By using multi-head attention, the model can simultaneously focus on information from different representation subspaces at various positions. This is not possible with a single attention head, as averaging would inhibit it. We assume the multi-headed attention mechanism of the H-heads is defined as follows.

$$\begin{aligned} head_h &= Attention(Q_h, K_h, V_h), \\ Q_h = QW_q^h, K_h &= KW_k^h, V_h = VW_v^h, h = 1, \cdots, H. \end{aligned} \tag{7}$$

Finally, the outputs of the H-heads are merged and then subjected to layer normalization to obtain the output of the attention layer, where W_q,W_k,W_v,W^O represent the weight matrices.

$$Attention(\mathcal{X}_{in}) = LayerNorm(Concat(head_1, \cdots, head_H)W^O). \tag{8}$$

Feed Forward Networks. In our encoder and decoder, each block contains not only an attention sublayer but also a fully connected feed forward network. Such network is applied after the attention layer and consists of two linear transformations and an activation function. We define a_r, W_r and b_r as learnable parameters, the activation function uses the LeakyReLU function, as follows:

$$FeedForward(\mathcal{X}_{res}) = max(a_r\mathcal{X}_{res}, \mathcal{X}_{res})W_r + b_r. \tag{9}$$

5 Evaluation

5.1 Dataset

The performance of HDCformer was validated on two real-world traffic flow datasets, PeMSD4 and PeMSD8. The datasets were collected from the California highway traffic flow data by the Caltrans Performance Measurement System (PeMS) [7], with data aggregated at 5-min intervals, i.e., 12 sample points per hour. Details of the two public datasets are given in Table 1.

Table 1. Details of the two public datasets.

Datasets	Nodes	Time Interval	Timesteps	Time Range
PeMSD4	307	5 min	16992	1/1/2018-2/28/2018
PeMSD8	170	5 min	17856	7/1/2016-8/31/2016

5.2 Baseline

We compared HDCformer with the following 7 representative and advanced baselines:

- **HA** [15]: History Average Model, which uses the average historical traffic flow of a road section within a certain time interval as the predicted value.
- **ARIMA** [12]: Autoregressive Integrated Moving Average Model, which treats the sequence as a random time series and approximates it using a mathematical model.
- **KNN** [8]: K-Nearest Neighbor Model, which finds neighbors in the historical database that match the current real-time observation data and uses a prediction algorithm to obtain the traffic prediction for the next moment.
- **SVR** [6]: Support Vector Regression Model, which is an application of support vector machine to regression problems.
- **LSTM** [30]: Long Short-Term Memory Model, which is a special type of RNN designed to solve the problem of gradient vanishing and gradient explosion during training of long sequences.
- **GRU** [9]: Gate Recurrent Unit Model, which is also a type of RNN that more efficiently solves the problem of long-term memory and gradient in backpropagation.
- **RPConvformer** [29]: A novel Transformer-based deep neural network for traffic flow prediction, which developed a fully convolutional embedding layer and used relative position encoding for linear mapping in multi-head attention.

5.3 Setting

Our HDCformer is implemented using Ubuntu 20.10, Python 3.7 and the deep learning framework TensorFlow 2.2.0. We conduct and evaluate all of our experiments on a server equipped with an 8-core Intel(R) Xeon(R) Platinum 8163 2.50GHz CPU and an NVIDIA Tesla T4 16GB GPU.

We split two datasets into training set, validation set, and test set in a ratio 6:2:2. We use the past day's historical data (288 timesteps) to predict the traffic flow data for the next hour. Further, the hyperparameters in the model are set as follows: the initial learning rate is set to 0.001 with Adam as the optimizer, the batchsize B is set to 32, the maximum epoch is 100, the hybrid dilated convolution layer has a convolution kernel of 7×7 with dilation rates of [1,2,5], and the number of encoders and decoders is set to 3 with 8 attention heads.

We use three commonly used metrics to assess the performance of HDCformer model: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE). The mathematical representations are as follows,

$$\begin{aligned} \text{MAE} &= \frac{1}{N}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right|, \\ \text{RMSE} &= \sqrt{\frac{1}{N}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}, \\ \text{MAPE} &= \frac{1}{N}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{y_i} \times 100\% \end{aligned} \tag{10}$$

where $\hat{y}_i$ represents the the predicted value, y_i represents the ground truth, N represents the number of samples in the test set.

5.4 Performance Comparison

Table 2 shows the comparison results of HDCformer and 7 baseline methods for traffic flow prediction on two public datasets. The results in bold indicate the best, and the results underlined are the second best. As can be seen, our proposed method outperforms all other baselines on all three indicators, especially compared to the most important baseline RPConvformer, HDCformer improves the prediction accuracy on PEMSD4 and PEMSD8 datasets by an average of 8.12% and 3.00%, respectively. Based on the experimental results, we can draw the following conclusions:

(1) Classical machine learning methods such as KNN and SVR models do not fully utilize sequence information, ignore the relevance of local trends in traffic flow sequences, and are easily affected by local outliers, resulting in poor generalization ability of the model.
(2) The serial attribute of deep learning-based methods such as LSTM and GRU models leads to the continuous transmission and amplification of errors in the recursive process.
(3) The RPConvformer model, based on one-dimensional convolution for sequence embedding, has a receptive field limited by the size of the convolution kernel, and the local sequence features learned cannot well represent trend relevance.

In summary, HDCformer combines improved strategies of hybrid dilated convolutional networks and evolutionary mechanisms of residual connections with the transformer framework for traffic flow prediction. Experimental results validate the superiority of the model.

5.5 Ablation Experiments

In order to further study the effectiveness of the HDCformer model in traffic flow prediction tasks, we designed the following variants for the data embedding module based on hybrid dilated convolution and the evolved transformer network.

Table 2. Predictive performance on the PEMSD4 and PEMSD8 datasets.

Model	PeMSD4			PeMSD8		
	MAE	**RMSE**	**MAPE(%)**	**MAE**	**RMSE**	**MAPE(%)**
HA	47.17	70.14	22.98	28.46	36.3	25.92
ARIMA	64.34	84.20	36.93	30.00	38.22	27.76
KNN	52.86	72.25	26.10	22.49	29.85	18.65
SVR	53.81	71.48	29.02	21.54	27.55	19.50
LSTM	38.50	52.06	19.23	19.75	25.96	16.96
GRU	39.78	52.25	22.52	20.19	26.68	17.01
RPConvformer	35.8	47.5	17.51	16.15	21.08	11.02
HDCformer(ours)	**32.80**	**43.60**	**16.15**	**15.71**	**20.54**	**10.61**

- HDCformer(no-dilated-conv): This variant removes the dilated convolution in the data embedding layer.
- HDCformer(same-dilated-rate): This variant sets the dilation rate of the three layers of dilated convolution to r=2 uniformly to evaluate the impact of gridding effects on the model.
- HDCformer(no-skip-connection): This variant removes the residual connections added between adjacent encoder layers, i.e., it degrades to the original transformer's encoder stacking method.

The predictive performance of HDCformer and the above three variant models on the PEMSD4 and PEMSD8 datasets were compared, and the ablation study results are shown in Fig. 3. By analyzing the experimental results, we can draw the following conclusions:

(1) HDCformer has a significant improvement in predictive performance compared to HDCformer(no-dilated-conv), which indicates that using dilated convolution for data embedding is effective in capturing local correlations in sequences.
(2) The performance of HDCformer(same-dilated-rate) is worse than that of HDCformer, which indicates that the gridding effect of dilated convolution causes the model to learn incomplete local sequence features and obtain defective trend information.

(3) The performance of HDCformer(no-skip-connection) is worse than that of HDCformer because this variant ignores the inheritance of attention patterns between layers and the attention scores of each layer are calculated independently.

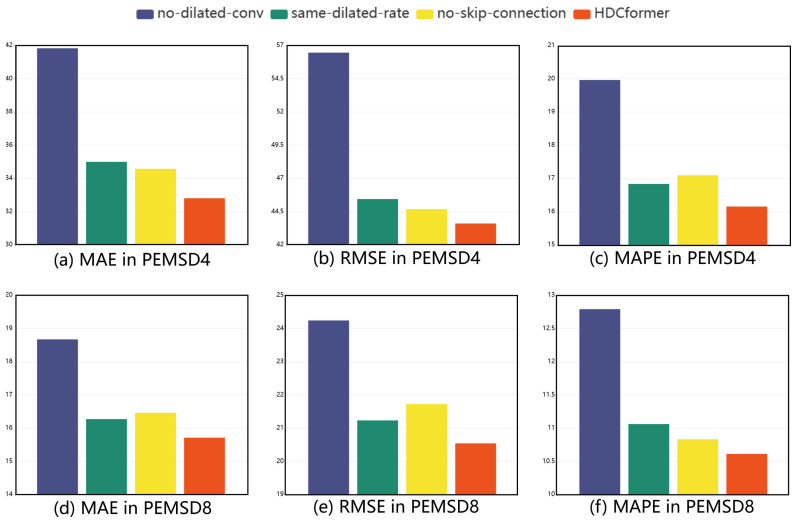

Fig. 3. Ablation study on the PEMSD4 and PEMSD8 datasets

6 Conclusion

In this paper, we propose An Evolving Transformer Network based on Hybrid Dilated Convolution for Traffic Flow Prediction. Specifically, we developed a data embedding layer based on hybrid dilated convolution by stacking dilated convolutions with different dilation rates to capture the local correlations in traffic flow time series and enable the model to learn the local trend of the sequence. We further connected the attention scores of adjacent layers in the original transformer to capture the evolution trend of attention patterns between different layers. Compared to the best baseline, our model improved by 8.12% and 3.00% on the two datasets respectively, demonstrating the superior performance of the HDCformer model. Ablation experiment results show that using a data embedding layer constructed with dilated convolution significantly improves the prediction accuracy of the model. Adding residual connections between adjacent attention modules also achieves the purpose of improving performance. As future work, we will further study the spatio-temporal correlation of traffic flow in road network structures, building on HDCformer and capturing the spatial dependence of traffic flows using, for example, spatial self-attention mechanisms and spatial graph modelling.

Acknowledgments. This work was supported by the Key-Area Research and Development Program of Guangzhou City (No. 202206030009).

References

1. Ba, J., Kiros, J., Hinton, G.: Layer normalization (2016)
2. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate (2014)
3. Bai, S., Kolter, J., Koltun, V.: An empirical evaluation of generic convolutional and recurrent networks for sequence modeling (2018)
4. Borovykh, A., Bohte, S., Oosterlee, C.W.: Dilated convolutional neural networks for time series forecasting. J. Comput. Finan. (2018). https://doi.org/10.21314/jcf.2019.358
5. Cai, L., Janowicz, K., Mai, G., Yan, B., Zhu, R.: Traffic transformer: capturing the continuity and periodicity of time series for traffic forecasting. Trans. GIS **24**, 736–755 (2020)
6. Castro-Neto, M., Jeong, Y.S., Jeong, M.K., Han, L.D.: Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions. Expert Syst. Appl. **36**(3), 6164–6173 (2008). https://doi.org/10.1016/j.eswa.2008.07.069
7. Chen, C., Petty, K., Skabardonis, A., Varaiya, P., Jia, Z.: Freeway performance measurement system: mining loop detector data. Transp. Res. Rec.: J. Transp. Res. Board **1748**, 96–102 (2007). https://doi.org/10.3141/1748-12
8. Cheng, S., Lu, F., Peng, P., Wu, S.: Short-term traffic forecasting: an adaptive ST-KNN model that considers spatial heterogeneity. Comput. Environ. Urban Syst. **71**, 186–198 (2018). https://doi.org/10.1016/j.compenvurbsys.2018.05.009
9. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling (2014)
10. Cui, Z., Ke, R., Wang, Y.: Deep bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction (2018)
11. Franceschi, J.Y., Dieuleveut, A., Jaggi, M.: Unsupervised scalable representation learning for multivariate time series (2019)
12. Isufi, E., Loukas, A., Simonetto, A., Leus, G.: Autoregressive moving average graph filtering. IEEE Trans. Sign. Proc. **65**(2), 274–288 (2016). https://doi.org/10.1109/tsp.2016.2614793
13. Jiang, W.: Internet traffic prediction with deep neural networks. Internet Technol. Lett. **5**(2), e314 (2021). https://doi.org/10.1002/itl2.314
14. Jiang, W., Luo, J.: Graph neural network for traffic forecasting: a survey. Expert Syst. Appl. **207**, 117921 (2022). https://doi.org/10.1016/j.eswa.2022.117921
15. Kaysi, I., Ben-Akiva, M., Koutsopoulos, H.: Integrated approach to vehicle routing and congestion prediction for real-time driver guidance (1993)
16. Lei, Y., et al.: The development of traffic flow prediction based on deep learning: a literature review. In: 2022 7th International Conference on Computer and Communication Systems (ICCCS) (2022). https://doi.org/10.1109/icccs55155.2022.9845878
17. Li, S., et al.: Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting (2019)
18. Liang, Y., Ke, S., Zhang, J., Yi, X., Zheng, Y.: GeoMAN: multi-level attention networks for geo-sensory time series prediction. In: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (2018). https://doi.org/10.24963/ijcai.2018/476

19. van Lint, J.W.C., Hoogendoorn, S.P., van Zuylen, H.J.: Freeway travel time prediction with state-space neural networks: modeling state-space dynamics with recurrent neural networks. Transp. Res. Rec.: J. Transp. Res. Board **1811**(1), 30–39 (2007). https://doi.org/10.3141/1811-04
20. Liu, S., et al.: Pyraformer: low-complexity pyramidal attention for long-range time series modeling and forecasting (2021)
21. Ma, C., Dai, G., Zhou, J.: Short-term traffic flow prediction for urban road sections based on time series analysis and LSTM_BILSTM method. IEEE Trans. Intell. Transp. Syst. **23**, 5615–5624 (2021). https://doi.org/10.1109/tits.2021.3055258
22. Qin, Y., Song, D., Chen, H., Cheng, W., Jiang, G., Cottrell, G.W.: A dual-stage attention-based recurrent neural network for time series prediction. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (2017). https://doi.org/10.24963/ijcai.2017/366
23. Shabani, A., Abdi, A., Meng, L., Sylvain, T.: Scaleformer: iterative multi-scale refining transformers for time series forecasting (2022)
24. Shih, S.Y., Sun, F.K., Lee, H.Y.: Temporal pattern attention for multivariate time series forecasting. Mach. Learn. **108**, 1421–1441 (2019). https://doi.org/10.1007/s10994-019-05815-0
25. Tedjopurnomo, D.A., Bao, Z., Zheng, B., Choudhury, F.M., Qin, A.K.: A survey on modern deep neural network for traffic prediction: trends, methods and challenges. IEEE Trans. Knowl. Data Eng. **34**(4), 1544–1561 (2022). https://doi.org/10.1109/TKDE.2020.3001195
26. Vaswani, A., et al.: Attention is all you need (2017)
27. Wang, P., et al.: Understanding convolution for semantic segmentation (2018)
28. Wang, Y., et al.: Convolution-enhanced evolving attention networks (2022)
29. Wen, Y., Xu, P., Li, Z., Xu, W., Wang, X.: RPConvformer: a novel transformer-based deep neural networks for traffic flow prediction (2023)
30. Williams, R., Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**, 1735–1780 (1997)
31. Wu, H., Xu, J., Wang, J., Long, M.: Autoformer: decomposition transformers with auto-correlation for long-term series forecasting (2021)
32. Wu, Z., Pan, S., Long, G., Jiang, J., Zhang, C.: Graph WaveNet for deep spatial-temporal graph modeling. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (2019). https://doi.org/10.24963/ijcai.2019/264
33. Xue, H., Salim, F.D.: TERMCast: temporal relation modeling for effective urban flow forecasting. In: Karlapalem, K., et al. (eds.) Advances in Knowledge Discovery and Data Mining. Lecture Notes in Computer Science(), vol. 12712, pp. 741–753. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-75762-5_58
34. Yi, S., Ju, J., Yoon, M.K., Choi, J.: Grouped convolutional neural networks for multivariate time series (2017)
35. Zhang, Y., Yan, J.: Crossformer: transformer utilizing cross-dimension dependency for multivariate time series forecasting (2022)
36. Zhou, H., et al.: Informer: beyond efficient transformer for long sequence time-series forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 11106–11115 (2022). https://doi.org/10.1609/aaai.v35i12.17325