IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 5 February 2024, accepted 22 March 2024, date of publication 27 March 2024, date of current version 4 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3381987

RESEARCH ARTICLE

MTESformer: Multi-Scale Temporal and Enhance Spatial Transformer for Traffic Flow Prediction

XINHUA DONG, WANBO ZHAO, HONGMU HAN, ZHANYI ZHU, AND HUI ZHANG
School of Computer Science, Hubei University of Technology, Wuhan 430074, China

Corresponding author: Xinhua Dong (xhdong@hbut.edu.cn)

This work was supported in part by the Science and Technology Project of the Department of Transport of Hubei Province under Grant 2022-11-4-3; in part by the Key-Area Research and Development Program of Guangdong Province under Grant 2020B1111420002; in part by the Key-Area Research and Development Program of Hubei Province under Grant 2022BAA040; and in part by the Innovation Fund of Hubei University of Technology under Grant BSQD2019027, Grant BSQD2019020, and Grant BSQD2016019.

ABSTRACT Traffic flow prediction has become an important component of intelligent transportation systems. However, high-precision traffic flow prediction (especially long-term prediction) is still very challenging due to the complex spatial-temporal dependences of urban traffic data. In this paper, a novel Multi-scale Temporal and Enhance Spatial Transformer (MTESformer) model is proposed to capture complex dynamic spatial-temporal dependencies. MTESformer provides a reasonable feature embedding of periodic characteristics of traffic; it can recognize different temporal feature patterns and capture long-term dependencies, and efficiently focuses on two different node-space dependencies (long-range and neighboring nodes dependencies). Specifically, we develop a special multi-scale convolution unit that unites temporal self-attention to capture a wider range of dynamic temporal dependencies from a multi-receptive field and identify different temporal feature patterns. Secondly, we design a novel Enhance Spatial Transformer module, which can better focus on the dynamic spatial dependencies among nodes by fusing their neighborhood information. Experimental results on the public transportation network datasets METR-LA, PEMS-BAY, PEMS04, and PEMS08 data show that our proposed method outperforms most of the baseline models and outperforms the state-of-the-art models in long-term prediction. (The MAE of 60min prediction of our model on METR-LA, PEMS-BAY dataset is 3.37, 1.87, and the MAPE is 9.62%, 4.35%, respectively, and all of them outperform the PDFormer on PEMS04 and PEMS08 datasets.)

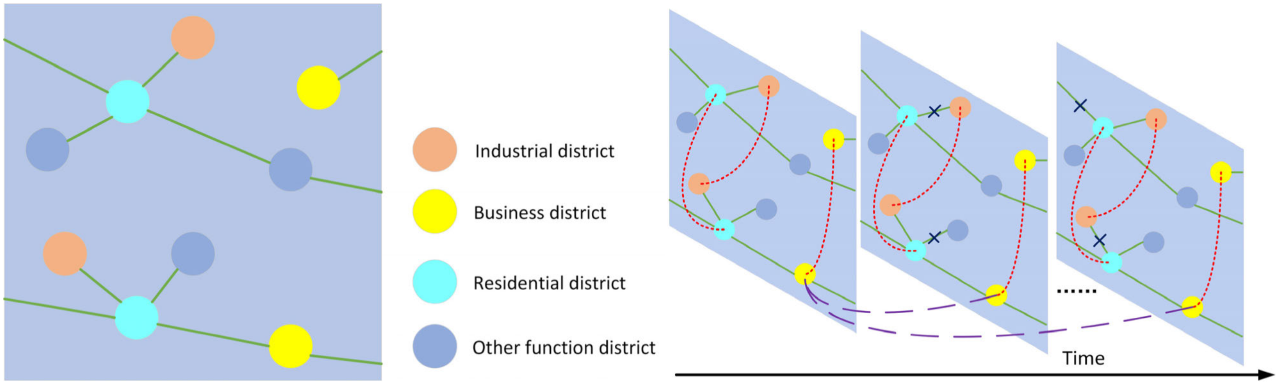
INDEX TERMS Long-term traffic flow prediction, multi-scale convolution, spatial-temporal dependency, transformer.

I. INTRODUCTION

With the rapid development of the economy and the acceleration of urbanization, the traffic flow in the transportation road network is growing, making the traffic congestion problem increasingly serious, and the problem of accurate traffic flow prediction becomes more and more urgent. Traffic flow forecasting is the process of using various historical data from the past in a given area to predict the volume of traffic in a given time period in the future. Traffic flow prediction [1], as a core technology of Intelligent Transportation Systems (ITS), has been widely studied, including traditional statistical methods and machine learning and deep learning methods. Accurate traffic flow forecasts can be used for a variety of transportation-related applications, including route planning, vehicle scheduling, and congestion mitigation, and they play an important role in traffic management and planning in cities, where they can help to alleviate traffic congestion and improve transportation services.

Due to the influence of various factors, traffic flow changes in a traffic road network show a complex and highly dynamic pattern of changes. Nodes (each recording point) in a traffic road network are affected by their neighboring nodes, which is evident due to various potential factors (e.g., traffic accidents, road closures, congestion, etc.). On the other hand,

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Wang.



(a)Schematic diagram of the road network structure.

(b) Complex dynamic spatial-temporal dependencies.

FIGURE 1. A simple simulation of the dynamic spatial-temporal dependencies in the road network by means of a static road network map and a dynamic spatial-temporal transformation map.

two nodes with similar urban functions may have spatial dependencies over long distances, even if their locations are far apart. And the temporal dependency of the same node may span multiple time steps. As demonstrated in Fig.1. In Fig.1-(a), the circles represent nodes in the road network, and the green lines represent actual roads, which enumerate nodes with different urban functions and show the connectivity between nodes. Fig.1-(b) shows the dynamic spatial-temporal dependency. Spatial dependency: red dashed lines connect nodes with similar urban function, despite their distance, due to the fact that they share the same traffic data pattern (e.g., industrial areas can be congested at the same time due to morning and evening peaks); the fork sign indicates that the connectivity of the road may be affected by potential factors (e.g., car accidents, congestion, etc.). Temporal dependency: Purple dashed lines indicate temporal dependencies of nodes that may span multiple time steps.

How to effectively capture complex and dynamic spatial-temporal dependencies and model traffic data is the core challenge in traffic flow prediction [2], and complex spatial-temporal interactions greatly increase the difficulty of the traffic prediction task. In recent years, a large number of deep learning models have been used to solve such spatial-temporal problems. Spatial correlations have been extensively explored using convolutional neural networks [3], [4], and recurrent neural networks (RNNs) have been widely used to learn temporal dynamics [5], [6]. However, CNN methods are suitable for capturing local spatial correlations in regular spatial grids, but are ineffective for traffic prediction in road networks with a variety of long-distance spatial correlations and belonging to an off-grid structure. Sequence learning methods for RNNs require iterative training, which progressively introduces error accumulation and incurs additional training time, and encounters the problem of gradient explosion or gradient vanishing when capturing long-term time dependencies [7], [8], [9]. Later, Given the superior performance of graph neural networks (GNNs) in processing graph-structured data, and given the fact that spatial-temporal data often comes with an underlying graph structure, GNN-based models were widely used to explore spatial-temporal properties [10], [11], [12]. Despite its natural advantages for handling data with non-Euclidean spatial structure, GNN-based approaches still encounter constraints in traffic forecasting. First, spatial dependencies in road networks are highly dynamic and such dependencies change over time; second, existing methods are usually locally designed and cannot capture remote dependencies. And these related models, which lack the ability to capture long-term dependencies, are still deficient in long-term prediction.

In addition, traffic flow data, there are complex variations and irregular dynamic similarities. Over long periods, there are similar daily patterns of morning and evening peak flow changes, as well as differences in traffic patterns between weekdays and days off. At the micro level, a large number of stochastic factors (e.g., different driving habits, roadway closures, etc.) make the flow changes highly dynamic, as shown in Fig.2. However, most approaches e.g [13] and [14], they lack a simultaneous focus on short-term and long-term characteristics.

To address the above problems, we propose a novel neural network framework for traffic flow prediction Multi-scale Temporal and Enhance Spatial Transformer (MTESformer). The main contributions of this paper can be summarized as follows:

(1) We propose the MTESformer model based on spatial-temporal self-attention for accurate prediction. Our method solves the problems of dynamic, long-range and long-term prediction with low accuracy.

(2) We design the Enhance Spatial Transformer, which fuses the neighborhood information of nodes into spatial self-attention, and can focus on both long-range dependencies and neighboring nodes that have more influence on them.

(3) We design a concise and effective multi-scale convolution unit (MSCU), which can discover temporal feature patterns at different scales through a multi-scale convolution

kernel, and further combine with temporal self-attention to mine more hidden temporal dependencies.

(4) Extensive experiments on real four datasets show that our model outperforms most of the baseline models and outperforms the state-of-the-art models in long-term prediction.

II. RELATED WORK

A. TRAFFIC FORECASTING

As a key functional component of intelligent transportation systems, traffic prediction has been widely studied and applied in various fields. Earlier, historical average (HA) [15], [16], traditional time series method ARIMA [15], machine learning models support vector regression(SVR) [17],vector autoregressive model (VAR) [18] and k-nearest neighbor (KNN) [19]were used for traffic prediction. However, these methods are strongly hypothetical and do not take into account the effect of spatial dependencies on the prediction, and are ineffective in dealing with traffic flow data with spatial-temporal dynamic complexity.

B. DEEP LEARNING METHOD

Deep learning models have become increasingly popular in solving traffic flow prediction problems. Considering that convolutional neural networks (CNNs) have demonstrated strong feature extraction capabilities in numerous applications, CNNs have been used to extract grid-based spatial dependencies [20], [21], [22]. However, the CNN method is suitable for capturing local spatial correlations in regular spatial grids, and for road networks with various long-range spatial correlations and belonging to off-grid structures, CNNs still have great limitations in capturing spatial dependencies. RNN models are a special approach to deal with time series, RNN-based models have been widely used in traffic prediction tasks in ITS [23], their variants Gated Recurrent Units (GRUs) [24] and Long Short-Term Memory Networks (LSTMs) [25]are used to simulate temporal dependencies in traffic prediction [26], [27]. In addition, Temporal Convolutional Networks TCN [28], a neural network structure based on causal convolution, have also been widely used to explore temporal dependencies due to its speed, small number of parameters, and structural soundness [29]. With the remarkable achievements of graph neural networks (GNNs) in the graph field, GNNs can be well adapted to the structure of road networks and have good performance [30], GNNs are widely used for traffic prediction [31], [32]. In addition, some works have used attention mechanisms to capture spatial-temporal dependencies due to their efficiency and flexibility [33], [34].

Currently, many works are based on the above approach to construct models to capture spatial-temporal dependencies and have achieved good results. Recurrent Neural Networks (RNN) have gained wide application in capturing temporal features due to their unique structure [35], which has a natural advantage in processing sequential data. Hussain et al. [36] used a stacked model of Bi-LSTM and GRU for traffic prediction, but its effectiveness still needs to be improved as it does not consider spatial dependency. Dynamic graph convolutional recurrent network (DGCRN) [37] constructs a dynamic adjacency matrix of node similarities and fuses GNN and RNN to capture spatial-temporal dependencies. Dual spatial convolution gated recurrent unit (DSC-GRU) [38] uses a DSC unit, which models global spatial dependencies by adding inter-node correlation coefficients on top of a static graph to generate a new dependency graph, and embeds the DSC unit in a GRU. Spatial-temporal gated recurrent unit (GCST-GRU) [39] uses GRU to capture temporal dependencies, and in terms of spatial modeling, explores spatially dependent optima in k-hop neighborhoods based on GCN, and explores regularization to improve the loss function. MGCN-WOALSTM [40] established MGCN (Multi-channel Graph Convolutional Neural network), which utilizes the self-attention mechanism to adjust the spatial correlation on different dimensions of the traffic flow, constructs LSTM to obtain temporal features, and introduces WOA (Whale Optimization Algorithm) to find the globally optimal combination of the parameters of the LSTM network. Spatial-temporal residual graph convolutional network (STRGCN) [41] optimizes serial GCN into DFRGCN with residual connectivity and capable of parallel computation, and learns historical temporal information between traffic streams by using an attention-based mechanism of bi-directional gated recurrent unit (ABi-GRU). Spatial-temporal dynamic graph convolutional neural network (STDGCN) [42] is a joint prediction model based on GCN and GRU, which performs cosine similarity computation on the nodes to generate dynamic neighbor matrix, and encodes and fuses the input features and node information. Another class of model components that captures time dependence is represented by TCN [43] based on 1D CNNs. Graph WaveNet [10] uses dilated causal convolution [13] to capture temporal dependencies, expands the receptive field by varying the size of the convolution kernel, and introduces an adaptive matrix, which is learned through node embeddings to capture hidden spatial dependencies in the data. Spatial-temporal graph attention network (STGAT) [44] constructed ST-Block by Gated Temporal Convolution Layer (GTCN) and graph attention layer (GAT) [45] and captured potential and existing spatial dependencies using dual path architecture. Progressive graph convolutional network (PGCN) [46] also uses dilated causal convolution and GCN together to capture spatial-temporal dependencies, but it uses an adjacency matrix measured by the cosine similarity of the node signals. Graph self-attention WaveNet (G-SWaN) [47] uses SGT (spatial graph transformer) instead of GCN on the basis of Graph WaveNet [10]and SGT can adjust the adaptive neighbor matrix and the true neighbor matrix.

It can be seen that the classical spatial-temporal graphical models (STGNN) have been widely used in the field of traffic flow prediction, but due to their own modeling constructs, these models are not as effective as they should be in long-term flow prediction. RNN models have obvious limitations for long-term prediction due to their iteration-based approach to time-dependent simulation; TCN uses one-dimensional convolutional kernel to capture the time dependency, using the right size of convolutional kernel can effectively capture the local time dependency, and in short-term prediction, better results have been achieved, such as (Graph WaveNet, MTGNN, and other models),but TCN still can't solve the long-term time dependency even if larger convolutional kernel is used. For spatial dependencies, many models explore hidden spatial dependencies other than the static adjacency matrix, DGCRN, PGCN and STDGCN generate new adjacency matrices by calculating node similarity, and Graph WaveNet captures hidden spatial dependencies using node-embedding dictionaries. These models have been extended in terms of spatial dependence to make up for the shortcomings of static adjacency matrices, and certain effect enhancements have been achieved. However, these methods still have defects: this spatial dependence is basically determined at the end of model training, and the dependence is unchangeable in the subsequent prediction process. Whereas the dependencies between nodes change over time, the node dependencies of these traffic flows are highly dynamic.

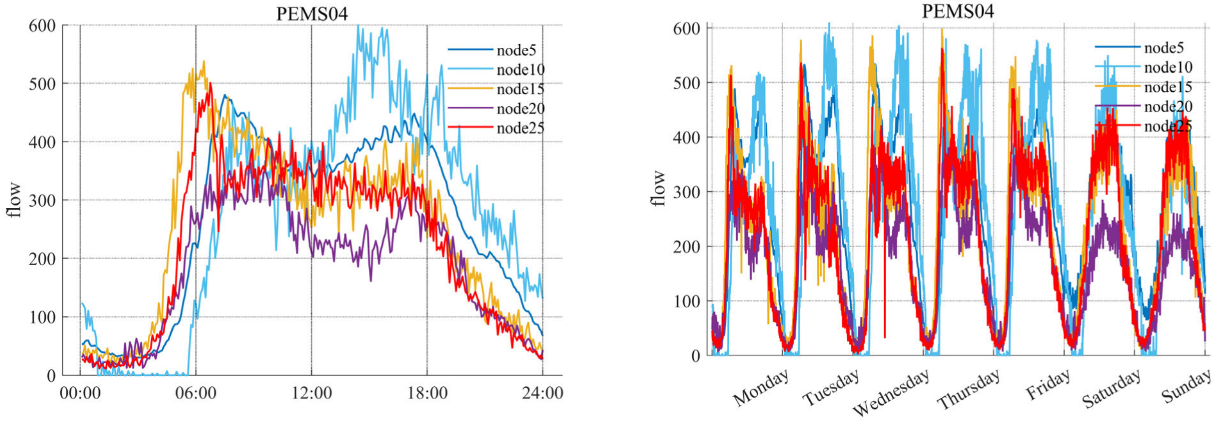
C. TRANSFORMER

Transformer [48] is a neural network model based on the self-attention mechanism. Transformer has demonstrated its effectiveness in natural language processing tasks and has gained extensive adoption in natural language processing [49] and other sequence modeling tasks. Transformer has also been successfully applied in computer vision tasks [50], [51]. Transformer is becoming increasingly popular due to its superior performance.

In addition, the attention mechanism, as a core part of Transformer, has been widely studied in the field of traffic flow prediction based on the self-attention variant model due to its effectiveness in capturing spatial-temporal dependencies [52]. Attention-based models can be dynamically adapted to capture spatial-temporal dependencies with real-time data, which is somewhat superior to STGNN models. Attention-based models have been widely used for traffic flow prediction, and they are more advantageous than STGNN models in long-term prediction with good results. Attention based spatial-temporal graph convolutional network (ASTGCN) [53] and multi-component attention graph convolutional network (MCAGCN) [54] extract information from traffic flow data of different time periods in three different period components, each of which is augmented with spatial-temporal convolution using attention mechanism. Spatial-temporal transformer networks (STTN) [55] capture temporal dependencies using temporal self-attention and spatial dependencies using spatial self-attention fused with GCN, respectively. Graph multi-attention network (GMAN) [33] fuses temporal attention with spatial attention to form STAtt Block and employs an encoder-decoder structure. Fast pure transformer network (FPTN) [56] also achieved good results using only self-attention for spatial dependence modeling and sensible embedding of time-periodic features. Adaptive graph spatial-temporal transformer network (ASTTN) [57] uses localized multiple self-attention, stacks multiple spatial-temporal attention layers, restricts attention to spatially adjacent nodes, and introduces adaptive graphs to capture hidden spatial-temporal dependencies. Dynamic spatial-temporal aware graph neural network (DSTAGNN) [58] proposes dynamic spatial-temporal aware graph to enhance the node association and M-GTU module to capture the dynamic information and improve the traditional spatial-temporal attention module. Spatial-temporal attention fusion dynamic graph convolution network (AFDGCN) [59] extends the fully-connected operations in GRU to GCN to form dynamic graph convolutional recurrent network (DGCGRU) and further uses temporal self-attention and GAT for spatial-temporal information enhancement.

For self-attention based models, ASTGCN and MCAGCN use different traffic cycle features for fusion, but this approach, using a three-path framework, greatly increases the computational complexity; STTNs and GMANs are modeled using only ordinary spatial-temporal attention, lacking further exploration of spatial-temporal dependence, and their effectiveness remains to be improved. ASTTN and FPTN focus the model on spatial dependence and achieve good results, indicating the effectiveness of exploring the spatial dependence of nodes for the improvement of traffic flow prediction. DSTAGNN incorporates graph convolution based on Chebyshev polynomial approximation into attention and uses M-GTU (multi-scale Gated Tanh Unit) to obtain extensive dynamic time dependence. AFDGCN uses temporal self-attention to augment DGC-GRU with GCN modules and GAT to attend to neighboring nodes. Although DSTAGNN and AFDGCN take into account the effects of multi-scale temporal dependence and domain nodes, respectively, their models are overly complex and provide little enhancement.

The above models do not make reasonable use of the periodicity of traffic and do not pay attention to both the multi-scale temporal dependence and the complex dynamic spatial dependence of the nodes, leading to their poor prediction results. To address these issues, we propose MTESformer. MTESformer does not overcomplicate the model by using too many components, but at the same time solves these complex dependencies and achieves good results in long-term forecasting. MTESformer makes reasonable use of the periodicity of traffic for periodic feature embedding; the use of MSCU in conjunction with temporal self-attention allows for the discovery of temporal feature patterns at different scales and captures long term temporal dependencies; and the incorporation of three-hop adjacency matrices into improved spatial self-attention pays better attention to the dynamic spatial dependencies between nodes (long-distance spatial dependencies and the influence of neighboring nodes).



(a) Daily traffic variations at 5 nodes.

(b) Weekly traffic variation at 5 nodes.

FIGURE 2. 5 nodes in the PEMS04 dataset and their flow variations are shown for the daily and weekly cycles, respectively.

TABLE 1. Summary of datasets.

Dataset	Sensors(N)	Timesteps	Edges
METR-LA	207	34272	12688
PEMS-BAY	325	52116	22712
PEMS04	307	16992	2620
PEMS08	170	17856	3312

III. MATERIALS AND METHODS

A. MATERIALS

1) DATASET

Our model is validated on four real datasets. In addition, We set both the historical time steps and future time steps to 12. METR-LA and PEMS-BAY are divided into training, validation, and test sets in the ratio of 7:1:2. PEMS04 and PEMS08 are divided in the ratio of 6:2:2.

METR-LA, the collection site was Los Angeles County freeways. The dataset contains a total of 207 sensors and covers the time period from 3/1/2012 to 6/30/2012.

PEMS-BAY, the collection site is in the Bay Area, California. The dataset contains a total of 325 sensors with a time range of January 2017 through May 2017.

PEMS04, collected from CalTrans PeMS. The dataset contains a total of 307 sensors and covers the time period from January 2018 through February 2018.

PEMS08, collected from CalTrans PeMS. The dataset contains a total of 170 sensors and covers the time period from July 2016 through August 2016.

In addition, the spatial neighbor map of each dataset is constructed based on the actual road network. Where the number of #Edges is the sum of the number of edges in the multi-order matrix. A summary of the dataset information is shown in Table.1.

2) DATA PRE-PROCESSING

The time interval for flow data is 5 minutes. For the small amount of missing data, as in most models, we replace them with zeros and mask them in the calculations. And the data were normalized by z-score.

In this work, we represent a road network as a graph $G = (V, E)$, where V is the set of nodes containing N nodes and E is the set of edges representing the connectivity relationships between nodes. The adjacency matrix of the graph G is denoted by $A \in \mathbb{R}^{N\times N}$, where an element A_{ij} in the adjacency matrix A is equal to 1 if nodes v_i and v_j belong to V and there exist connecting edges $(v_i, vj) \in E$. We can denote the state of the traffic at any time step t as $X^t \in \mathbb{R}^{N\times C}$, where C denotes the type quantity of traffic parameters. In this study, our goal is to predict one parameter type, traffic flow $(C = 1)$

3) DATA AUGMENTATION

In order to better extract the hidden information of traffic features, we use the fully connected layer to obtain the traffic feature embedding $E_f \in \mathbb{R}^{T\times N\times D}$:

$$E_f = FC\left(X^{(t-T+1):t}\right) \tag{1}$$

where D is the embedded feature dimension of the model and $FC(\cdot)$ denotes the fully connected layer.

In addition, considering that the changes in traffic flow are influenced by people's lifestyle and commuting habits, which have a cyclical nature on a large scale, as shown in Fig.2, the daily flow changes and weekly flow changes are demonstrated for five nodes in the PEMS04 dataset. During the day, a significant increase in traffic occurs during the morning and evening peak hours, and during the week, the pattern of traffic variation varies again on weekdays and days off. Therefore, we introduce these two important features into our model as well. Specifically, we use the learnable day-of-week Embedding dictionary $T_d \in \mathbb{R}^{N_d\times D}$, and time-of-day Embedding dictionary $T_t \in \mathbb{R}^{N_t\times D}$ where, $N_d = 7$, $N_t = 288$ (number of timestamps per day). The day-of-week data $W_d \in R^T$ of the weekly time series at the corresponding moment of the flow is used as an index to extract the day-of-week Embedding $Z_d \in \mathbb{R}^{T\times D}$. The time-of-day data $W_t \in \mathbb{R}^T$ of the daily time

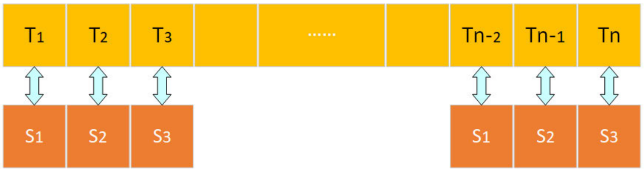


FIGURE 3. Schematic of a 1D CNN acting on a time series.

series at the corresponding moment of the flow is used as an index to extract the time-of-day Embedding $Z_t \in \mathbb{R}^{T \times D}$. Finally, the time-period embedding features $E_d \in \mathbb{R}^{T \times N \times D}$and $E_t \in \mathbb{R}^{T \times N \times D}$of the N nodes are obtained by broadcasting.

We add the above 3 embedding vectors to get the embedding feature input:

$$X = E_f + E_t + E_d \tag{2}$$

B. METHODS

1) CNN

One-dimensional CNNs use a similar approach to TCNs to process time series. As shown in Fig.3, a time series of length n is processed using a convolutional kernel of size 1*3. The 1*3 convolution kernel will be shifted along the time axis dimension, and the processing of the time series will result in a sequence of length (n-3+1). With different sizes of convolutional kernels, temporal information can be extracted at different scales.

2) ATTENTION

Multinomial self-attention can be attentive to the global scope, and it has been widely used to capture spatial-temporal dependencies. The core idea is to first perform self-attention by query and key to get the attention score, and then update the value. The operation of the ith header is as follows, query, key, and value are obtained by the projection of X, respectively:

$$XW_q^{(i)} = Q^{(i)}, XW_k^{(i)} = K^{(i)}, XW_v^{(i)} = V^{(i)} \tag{3}$$

The value is then updated by the scaled dot product function and softmax operation to compute the attention score $A^{(i)}$:

$$A^{(i)} = Softmax\left(\frac{Q^{(i)}K^{(i)^T}}{\sqrt{d}}\right) \tag{4}$$

$$Att\left(Q^{(i)}, K^{(i)}, V^{(i)}\right) = A^{(i)}V^{(i)} \tag{5}$$

where d is the number of feature dimensions of the ith header, then expand the single header into a multi-header by doing the following:

$$O^{(i)} = Att\left(Q^{(i)}, K^{(i)}, V^{(i)}\right) \tag{6}$$

$$O = Concat\left[O^{(1)}, O^{(2)}, \ldots, O^{(H)}\right] \tag{7}$$

IV. METHODOLOGY

A. SYSTEM MODEL

1) MULTI-SCALE TEMPORAL TRANSFORMER

Since the self-attention mechanism is unable to capture the temporal position information of the observed time series, we use position embedding to inject the "temporal position" information into the input time series. The temporal position embedding information $\hat{P}_t \in \mathbb{R}^{T \times N \times D}$ is learned through the learnable dictionary $P_t \in \mathbb{R}^{T \times D}$ and the broadcasting mechanism, The output X of the Feature Embedding Layer is then summed with $\hat{P}_t$ to get $X_t^{'}$:

$$X_t' = X + \hat{P}_t \tag{8}$$

By adopting the multi-head self-attention mechanism in combination with multi-scale convolution unit (MSCU), it is possible to pay attention to both short-term and long-term correlations in time series data, discover temporal feature patterns at different scales, and capture hidden dependencies at multiple scales. The self-attention mechanism is first applied along the time axis to capture complex temporal dependencies. For a multi-attention model with H attention heads, we define the following variables:

$$X_t W_q^{(i)} = Q_t^{(i)}, X_t W_k^{(i)} = K_t^{(i)}, X_t W_v^{(i)} = V_t^{(i)} \tag{9}$$

$$A_t^{(i)} = Softmax\left(\frac{Q_t^{(i)} K_t^{(i)}}{\sqrt{d}}\right) \tag{10}$$

$$Att\left(Q_t^{(i)}, K_t^{(i)}, V_t^{(i)}\right) = A_t^{(i)} V^{(i)} \tag{11}$$

where $X_t \in \mathbb{R}^{N \times T \times D}$ is obtained from the output X_t' of the Feature Embedding Layer by reshaping, and $Q_t^{(i)}$, $K_t^{(i)}$, $V_t^{(i)} \in \mathbb{R}^{N \times T \times d}$ are obtained by projection from the learnable parameters $W_{q,k,v}^{(i)} \in \mathbb{R}^{D \times d}(d = D/H)$, and then the self-attention score $A_t^{(i)} \in \mathbb{R}^{N \times T \times T}$ by introducing the scaled dot product function and softmax row-by-row normalization operation.

The above is the operation of the ith head in multiple self-attention, we extend to multiple heads:

$$O_t^i = Att\left(Q_t^{(i)}, K_t^{(i)}, V_t^{(i)}\right) \tag{12}$$

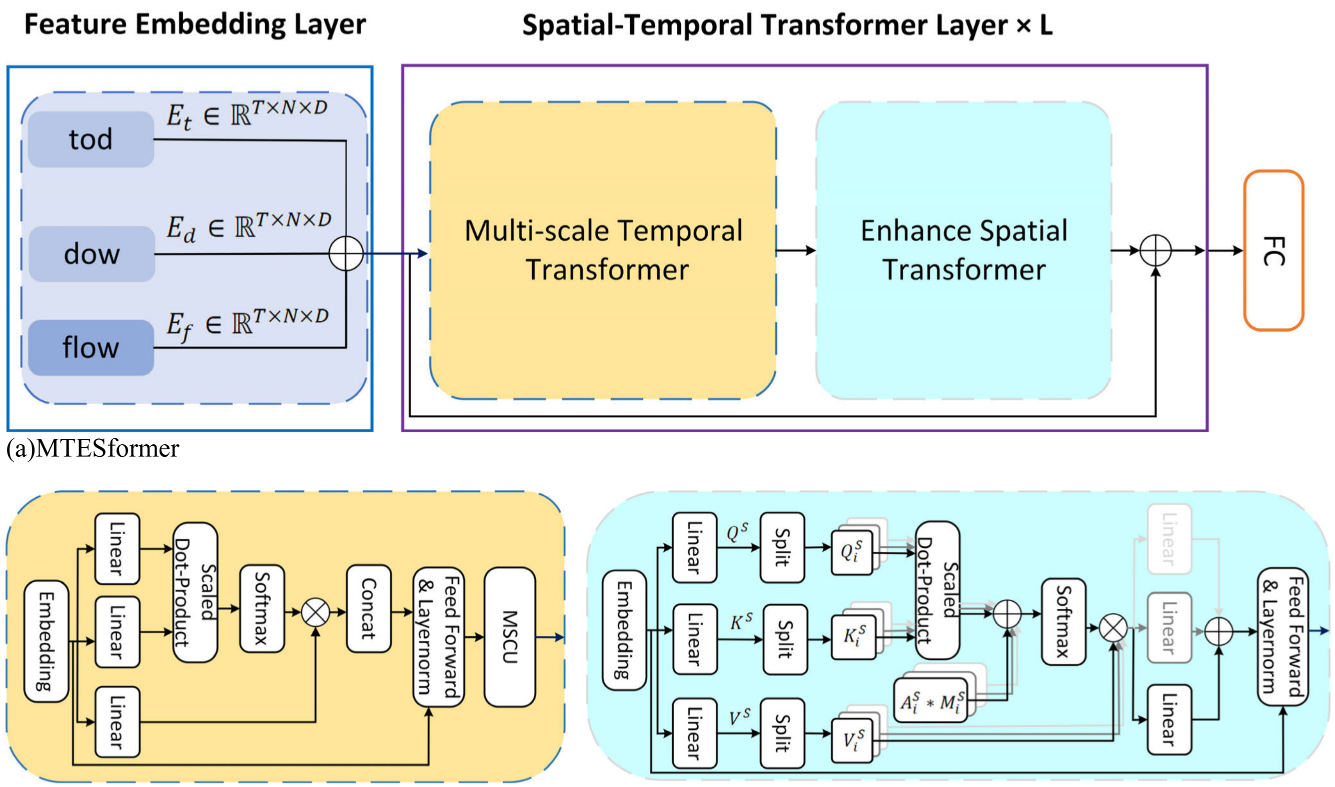
$$O_t = Concat\left[O_t^{(1)}, O_t^{(2)}, \ldots, O_t^{(H)}\right] \tag{13}$$

where $O_t^i \in \mathbb{R}^{N \times T \times d}$ denotes the output of the ith head of temporal self-attention, and $O_t \in \mathbb{R}^{N \times T \times D}$ is the output of H-heads of temporal self-attention spliced together in the last dimension, subsequently, O_t is sent to the Feed Forward & Layernorm layer after a linear transformation:

$$Y_t' = LayerNorm\left(Linear\left(O_t\right) + X_t\right) \tag{14}$$

$$Y_t = LayerNorm\left(FFN\left(Y_t'\right) + Y_t'\right) \tag{15}$$

where X_t is the input of multi-head self-attention, FFN is Feed Forward Network, and $Y_t \in \mathbb{R}^{N \times T \times D}$ is the output of Feed Forward & Layernorm layers.



(a)MTESformer

(b) Multi-scale Temporal Transformer

(c) Enhance Spatial Transformer

FIGURE 4. (a) Overall architecture of MTESformer, (b) and (c) show the specific details of each module in detail.

To further enhance the model's perception of dynamic temporal dependencies in the road network, we add a MSCU(multi-scale convolution unit) after the temporal multi-head self-attention, which discovers temporal feature patterns at different scales, specifically, as shown in Fig.5, the MSCU has four convolution kernels of different sizes, the convolution kernels are $\Gamma_1 \in \mathbb{R}^{1\times S_1\times D\times D}$, $\Gamma_2 \in \mathbb{R}^{1\times S_2\times D\times D}$, $\Gamma_3 \in \mathbb{R}^{1\times S_3\times D\times D}$, $\Gamma_4 \in \mathbb{R}^{1\times S_4\times D\times D}$, as well as the linear and ReLU activation layers, and residual connections are also set up for them. Where the kernel sizes are $1 \times S_1$, $1 \times S_2$, $1 \times S_3$, $1 \times S_4$. This is done as follows:

$$Z_t' = (Concat\,(\Gamma_1 \star Y_t, \Gamma_2 \star Y_t, \Gamma_3 \star Y_t, \Gamma_4 \star Y_t,)) \quad (16)$$

$$Z_t = Linear\left(ReLU\left(Linear\left(Z_t'\right)\right)\right) + Y_t \quad (17)$$

where $\star$ represents the convolution operation, $Y_t \in \mathbb{R}^{N\times T\times D}$ is the output of the temporal self-attention, which is obtained from the output of the temporal self-attention Y_t by the four convolution kernels respectively, after one-dimensional convolution on the timeline dimension, the dimensions on the timeline are $T-S_1+1, T-S_2+1, T-S_3+1, T-S_4+1$, and use the concat operation to splice them together in the time axis dimension, after the splice and after the first projection in the time axis dimension to get $Z_t' \in \mathbb{R}^{N\times Q\times D}$, and then map the number of dimensions in the time axis to T to get the output of the MSCU $Z_t \in \mathbb{R}^{N\times T\times D}$.

2) ENHANCE SPATIAL TRANSFORMER

Similar to Embedding in Multi-scale Temporal Transformer, we use positional embedding to inject information that distinguishes between different nodes into the output of Multi-scale Temporal Transformer, which we reshape to represent as $Z_s \in \mathbb{R}^{T\times N\times D}$, learn the spatial embedding information $\hat{P}_s \in \mathbb{R}^{T\times N\times D}$ by the learnable dictionary $P_s \in \mathbb{R}^{N\times D}$ and broadcasting mechanism, and then add Z_s and $\hat{P}_s$ to obtain $X_s \in \mathbb{R}^{T\times N\times D}$:

$$X_s = Z_s + \hat{P}_t \quad (18)$$

In capturing spatial dependencies, we focus on nodes near the center node and nodes that exhibit similar functions despite being distant from each other. Therefore, we develop a new module to capture dynamic spatial dependencies between nodes. We incorporate the adjacency matrix into the improved multi-head spatial self-attention to dynamically mine the hidden relationships between nodes based on historical information, and we denote the ith-hop adjacency matrix by $A^i \in \mathbb{R}^{N\times N}$ (if $A^i_{j,k}$ =1, it means that the k-node is the ith-hop neighbor of the j-node and the diagonal element is 1), for a multi-head spatial self-attention model with H multiple heads of attention model, we define the following variables:

$$X_s W_q^{(i)} = Q_s^{(i)}, X_s W_k^{(i)} = K_s^{(i)}, X_s W_v^{(i)} = V_s^{(i)} \quad (19)$$

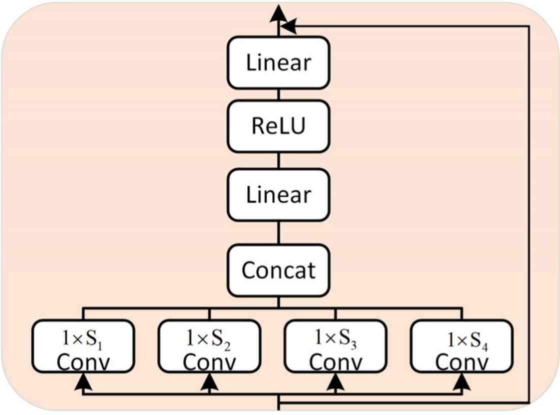


FIGURE 5. Multi-scale convolution unit.

$$P_s^{(i)} = Softmax\left(\frac{Q_s^{(i)}K_s^{(i)}}{\sqrt{d}} + A^i \odot M^i\right) \tag{20}$$

where $X_s \in \mathbb{R}^{T\times N\times D}$ is obtained from the output of the spatial Embedding in Enhance Spatial Transformer, $Q_s^{(i)}, K_s^{(i)}, V_s^{(i)} \in \mathbb{R}^{T\times N\times d}$ is obtained by projection from the learnable parameters $W_{q,k,v}^{(i)} \in R^{D\times d}(d = D/H)$, $M^i \in \mathbb{R}^{T\times N\times N}$ denotes the Mask matrix of the ith head, $\odot$ is the element-wise Hadamard product, M^i, A^i jointly adjust the attention scores for each head, and then the self-attention scores $P_s^{(i)} \in \mathbb{R}^{T\times N\times N}$ are obtained by introducing the scaling dot product function and softmax row-by-row normalization operation.

$$Att\left(Q_s^{(i)}, K_s^{(i)}, V_s^{(i)}\right) = P_s^{(i)}V_s^{(i)} \tag{21}$$

$$O_s = Sum\left[O_s^{(1)}W^{(1)}, O_s^{(2)}W^{(2)}, \ldots, O_s^{(H)}W^{(H)}\right] \tag{22}$$

where $O_s^i \in \mathbb{R}^{T\times N\times d}$ is the output of the ith head, $W^{(i)} \in \mathbb{R}^{T\times d\times D}$ is the feature transformation matrix, $O_s \in \mathbb{R}^{T\times N\times D}$ denotes the output of the multi-headed spatial self-attention layer, and subsequently, $O_s$is sent to the Feed Forward & Layernorm Layer after a linear transformation:

$$Y_s' = LayerNorm\left(Linear\left(O_s\right) + X_s\right) \tag{23}$$

$$Y_s = LayerNorm\left(FFN\left(Y_s'\right) + Y_s'\right) \tag{24}$$

where for FFN is Feed Forward Network and $Y_s \in \mathbb{R}^{T\times N\times D}$ is the output of Feed Forward & Layernorm layer.

3) PREDICT LAYER

After stacking multiple layers of Spatial-Temporal Transformer Layer, the output of the last layer is obtained as $O \in\mathbb{R}^{T\times N\times D}$, which is reshaped as $O' \in \mathbb{R}^{N\times TD}$, denoting features extracted from each of N nodes, and then passes through a layer of fully connected layers:

$$Y' = FC\left(O'\right) \tag{25}$$

where $Y' \in \mathbb{R}^{N\times MC}$, which is then reshaped to obtain the output of the final model $Y \in \mathbb{R}^{M\times N\times C}$, M represents the number of time steps being predicted, and $C = 1$ denotes the flow characteristics.

B. ARCHITECTURE AND WORKING

To cope with the dynamics of real-time data, we need to further study the dynamic characteristics of these dependencies in detail. Therefore, we propose a new spatial-temporal attention module which organically combines Multi-scale Temporal Transformer and Enhance Spatial Transformer to further enhance the extraction of dynamic spatial-temporal dependencies. The proposed MTESformer is shown in Fig.4-(a), MTESformer consists of Feature Embedding Layer and L layers of Spatial-Temporal Transformer Layer with residual connectivity, each Spatial-Temporal Transformer Layer in turn consists of a Multi-scale Temporal Transformer and an Enhance Spatial Transformer to jointly extract the spatial-temporal features, as well as a final prediction layer. The flow pattern of traffic data X is illustrated in Fig.4-(a). Firstly, X undergoes data augmentation through the Feature Embedding Layer, followed by the extraction of spatial-temporal dependencies using L layers of Spatial-Temporal Transformer. Finally, the Predict Layer is utilized to forecast the traffic data for future time intervals.

V. EXPERIMENTS

A. EXPERIMENTATION

Our experiments were performed in a Window environment using an Intel(R) Core(TM) i9-10900K CPU@3.70GHZ and a NVIDIA GeForce RTX 3090 GPU card. Indeed, our model is hyperparameter insensitive and widely adaptable with the following hyperparameters: the number of Spatial-Temporal Transformer Layers is 3, the embedding dimension D is 24, the number of heads of multi head self-attention H is 3, the sizes of the four one-dimensional convolutional kernels in the MSCU are 3, 5, 7, and 9, respectively, the intermediate layer timeline dimension Q in the MSCU is 64, hidden layer dimension in FFN is 256, history time step T and future time step M are both 12, Adma is used as the optimizer, the learning rate is decayed from 0.001, the batch size is 16, the number of iterations is 100, and an early stopping mechanism is used with the error of the validation set. The only difference is that we use Huber loss on PEMS04, PEMS08, use MAE loss on METR-LA, PEMS-BAY and remove the activation layer in MSCU. For a more detailed overview of the model's parameters, we present them in Table 2. We use MAE (Mean Absolute Error), MAPE (Mean Absolute Percentage Error), and RMSE (Root Mean Square Error) to validate our model.

$$MAE = \frac{1}{n}\sum\nolimits_{i=1}^{n}\left|\bar{y}_i - y_i\right| \tag{26}$$

$$RMSE = \sqrt{\frac{1}{n}\sum\nolimits_{i=1}^{n}\left(\bar{y}_i - y_i\right)^2} \tag{27}$$

$$MAPE = \frac{1}{n}\sum\nolimits_{i=1}^{n}\left|\frac{\bar{y}_i - y_i}{y_i}\right| \tag{28}$$

TABLE 2. Detailed experimental parameters.

	PEMS04	PEMS08	METR-LA	PEMS-BAY
learning rate	0.001			
weight decay	0.0005		0.0003	0.0001
gamma	0.1			
milestones	[35 55 70]	[45 85]	[35 55 70]	[20 30]
batchsize	16			
epochs	100			
[D Q]	[24 64]			
kernel_size	[3 5 7 9]			
num layers	3			
num heads	3			
loss	HuberLoss		MAELoss	

Among them, n represents the number of nodes, while $\bar{y}_i$ and y_i respectively represent the true value and predicted value for the i-th time step.

B. RESULTS AND ANALYSIS

1) RESULTS

In this study, our proposed model is compared to the following baselines.

SVR: Support Vector Regression, a classical time series regression task using linear support vector machines [17].

DCRNN: Diffusion convolutional recurrent neural network, merging diffusion map convolutional network and seq2seq for traffic flow prediction [8].

FC-LSTM: A special RNN model with fully connected LSTM layers [60].

AGCRN: Adaptive graph convolutional recurrent network combining GCN with GRU using learnable embedding of nodes in graph convolution [61].

STGCN: Combined graph convolutional and convolutional sequence learning layers for modeling [62].

GWNet: Will expand causal convolution and GCN to jointly capture spatial and temporal dependencies [10].

MTGNN: Capturing spatial-temporal dependencies using hybrid jump propagation layers, expanded initial layers, and graph learning modules [63].

GMAN: An attention-based model, with encoder-decoder architecture [33].

DSTAGNN: An improved spatial-temporal multi-attention model that incorporates GCN into spatial attention and uses multi-scale gated convolution [58].

AFDGCN: Augmentation of DGC-GRU with GCN module using temporal self-attention and attention to neighboring nodes using GAT [59].

PDFormer: A spatial and temporal self-attention model, with graph mask matrices and delay-aware feature transformation modules [64].

We conduct experiments on four datasets PEMS-BAY, METR-LA, PEMS04, and PEMS08, and the experimental results are shown in Table.3 and Table.4. MTESformer clearly outperforms most of the baseline models, especially in long-term prediction.

For the METR-LA dataset and the PEMS-BAY dataset, MTESformer's results are slightly worse compared to MTGNN and Graph WaveNet in the short-term predictions (15min and 30min), but in the 60min prediction, MTESformer performs much better than all baselines on all three metrics. On the METR-LA dataset, there is a significant decrease in both MAE and MAPE for MTESformer compared to GWNet and DCRNN; Compared to PDFormer, MTESformer predicted 0.35 and 1.29% lower MAE and MAPE at 60min, respectively; On the PEMS-BAY dataset, again, MTESformer's prediction at 60 min is significantly better than all models. On the PEMS04 and PEMS08 datasets, MTESformer outperforms all models on three evaluation metrics. And the advantage is more obvious on the PEMS08 dataset, where the MAE and MAPE of MTESformer are reduced by 1.18 and 1.2%, respectively, compared to GMAN, and by 1.69 and 1.28%, respectively, compared to DCRNN.

(1) Deep learning models significantly outperform traditional models such as SVR due to their strong assumptions about the data. FC-LSTM performs poorly because it does not consider spatial dependence.

(2) DCRNN, AGCRN are typical RNN-based methods for predicting spatial-temporal data. These models are limited by their lack of ability to maintain long-range temporal patterns due to their cyclic-based. Graph Wave Net combines GNN and Gated TCN with small inceptive field convolution kernel to form a spatial-temporal layer, which is more advantageous in short-term prediction. MTGNN uses a combination of a mix-hop propagation layer and a dilated inception layer for fusion, a strategy that has also yielded good results. These models rely heavily on CNN methods to capture temporal dependencies, but one-dimensional convolution is usually limited by the size of the receptive field and pays insufficient attention to long-range temporal information. Compared with these classical spatial-temporal fusion models with one-dimensional convolution or based on RNN, due to the structure of MTESformer based on the attentional mechanism, although it has shortcomings in short-term prediction, MTESformer has a huge advantage in long-term prediction, which is often more important.

(3) GMAN is modeled using spatial-temporal attention with an encoder-decoder architecture that lacks further exploration of spatial-temporal dependencies. PDFormer models local geographic neighborhoods and global semantic neighborhoods using a graph masking approach. DSTAGNN incorporates graph convolution based on Chebyshev polynomial approximation into attention and uses M-GTU (multi-scale Gated Tanh Unit) to obtain a wide range of dynamic temporal

TABLE 3. Performance on METR-LA and PEMS-BAY datasets.

Dataset	Model	15 min			30 min			60 min		
		MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
METR-LA	SVR	3.39	8.45	9.30%	5.05	10.87	12.10%	6.72	13.76	16.70%
	DCRNN	2.67	5.16	**6.86%**	3.12	6.27	8.42%	3.54	7.47	10.32%
	FC-LSTM	3.44	6.30	9.60%	3.77	7.23	10.09%	4.37	8.69	14.00%
	AGCRN	2.85	5.53	7.63%	3.20	6.25	9.00%	3.59	7.45	10.47%
	STGCN	2.75	5.29	7.10%	3.15	6.35	8.62%	3.60	7.43	10.35%
	GWNet	**2.69**	**5.15**	6.99%	3.08	6.22	8.47%	3.51	7.28	9.96%
	MTGNN	**2.69**	5.16	6.89%	3.05	**6.13**	**8.16%**	3.47	7.21	9.70%
	GMAN	2.80	5.55	7.41%	3.12	6.49	8.73%	3.44	7.35	10.07%
	PDFormer	2.83	5.45	7.77%	3.20	6.46	9.19%	3.62	7.47	10.91%
	MTESformer	2.73	5.34	7.10%	**3.03**	6.22	8.30%	**3.37**	**7.14**	**9.62%**
PEMS-BAY	SVR	1.85	3.59	3.80%	2.48	5.18	5.50%	3.28	7.08	8.00%
	DCRNN	1.31	2.76	2.73%	1.65	3.75	3.71%	1.97	4.60	4.68%
	FC-LSTM	2.05	4.19	4.80%	2.20	4.55	5.20%	2.37	4.96	5.70%
	AGCRN	1.35	2.88	2.91%	1.67	3.82	3.81%	1.94	4.50	4.55%
	STGCN	1.36	2.88	2.86%	1.70	3.84	3.79%	2.02	4.63	4.72%
	GWNet	**1.30**	**2.73**	**2.71%**	1.63	**3.73**	3.73%	1.99	4.60	4.71%
	MTGNN	1.33	2.80	2.81%	1.66	3.77	3.75%	1.95	4.50	4.62%
	GMAN	1.35	2.90	2.87%	1.65	3.82	3.74%	1.92	4.49	4.52%
	PDFormer	1.32	2.83	2.78%	1.64	3.79	3.71%	1.91	4.43	4.51%
	MTESformer	1.31	2.80	2.75%	**1.62**	**3.73**	**3.62%**	**1.87**	**4.38**	**4.35%**

TABLE 4. Performance on PEMS04 and PEMS08 datasets.

Datasets	PEMS04			PEMS08		
Metric	MAE	RMSE	MAPE	MAE	RMSE	MAPE
SVR	28.66	44.59	19.15%	23.25	36.15	14.71%
DCRNN	19.63	31.26	13.59%	15.22	24.17	10.21%
FC-LSTM	26.24	40.49	19.30%	22.20	33.06	15.02%
AGCRN	19.38	31.25	13.40%	15.32	24.41	10.03%
STGCN	19.57	31.38	13.44%	16.08	25.39	10.60%
GWNet	18.53	29.92	12.89%	14.40	23.39	9.21%
MTGNN	19.17	31.70	13.37%	15.18	24.24	10.20%
GMAN	19.14	31.60	13.19%	15.31	24.92	10.13%
DSTAGNN	19.30	31.46	12.70%	15.67	24.77	9.94%
AFDGCN	19.09	31.01	12.62%	15.02	24.37	9.68%
PDFormer	18.36	30.03	**12.00%**	13.58	23.41	9.05%
MTESformer	**18.23**	**29.78**	**12.00%**	**13.53**	**23.32**	**8.93%**

dependencies, but it fails to incorporate the periodic characterization of the traffic as well as to consider the interactions of domain nodes. AFDGCN uses temporal self-attention to augment DGC-GRU with GCN modules and GAT to attend to neighboring nodes. DSTAGNN and AFDGCN take into account the multi-scale time dependence and the influence of domain nodes, respectively, and their model components are overly complex, but the enhancement is minimal.

It is worth noting that MTESformer embeds traffic cycle features compared to models also based on self-attention; the use of MSCU in conjunction with temporal self-attention allows for the capture of multi-scale temporal dependencies and long temporal dependencies; and the incorporation of three-hop adjacency matrices into improved spatial self-attention allows for the effective focus on two different kinds of spatial dependencies. MTESformer simultaneously addresses the shortcomings of the above models. Instead of using overly complex components, the temporal features and effective spatial attention patterns are captured by more fine-grained modules, and MTESformer significantly outperforms both GMAN and PDFormer in both short and long-term forecasting on the METR-LA dataset and the PEMS-BAY dataset. On the PEMS04 and PEMS08 datasets, MTESformer was significantly better than the other baseline models, achieving the best results for all metrics.

(4) MTESformer outperforms most of the baseline models in the four real datasets, achieving optimal performance, especially in long-term prediction, with great advantages.

2) ANALYSIS

In order to assess the effectiveness of different components in MTESformer, we conduct a comparative study by contrasting MTESformer with the following variations. (1) w/o t_d_E: this variant removes time of day and day of week Embedding. (2) w/o E_S: this variant removes Enhance Spatial Transformer. (3) w/o MSCU: this variant removes MSCU (multi-scale convolution unit). (4) w/spatial_att: this variant replaces the Enhance Spatial Transformer with ordinary spatial self-attention.

The experimental results are shown in Table.5, and the specific prediction errors of each variant model on the PEMS04 and PEMS08 datasets are plotted in Fig.6, where w/o E_S has the highest error, and the error of w/o t_d_E is also higher compared to MTESformer, while the errors of w/o MSCU, w/spatial_att are slightly close to MTESformer but still have some gap.

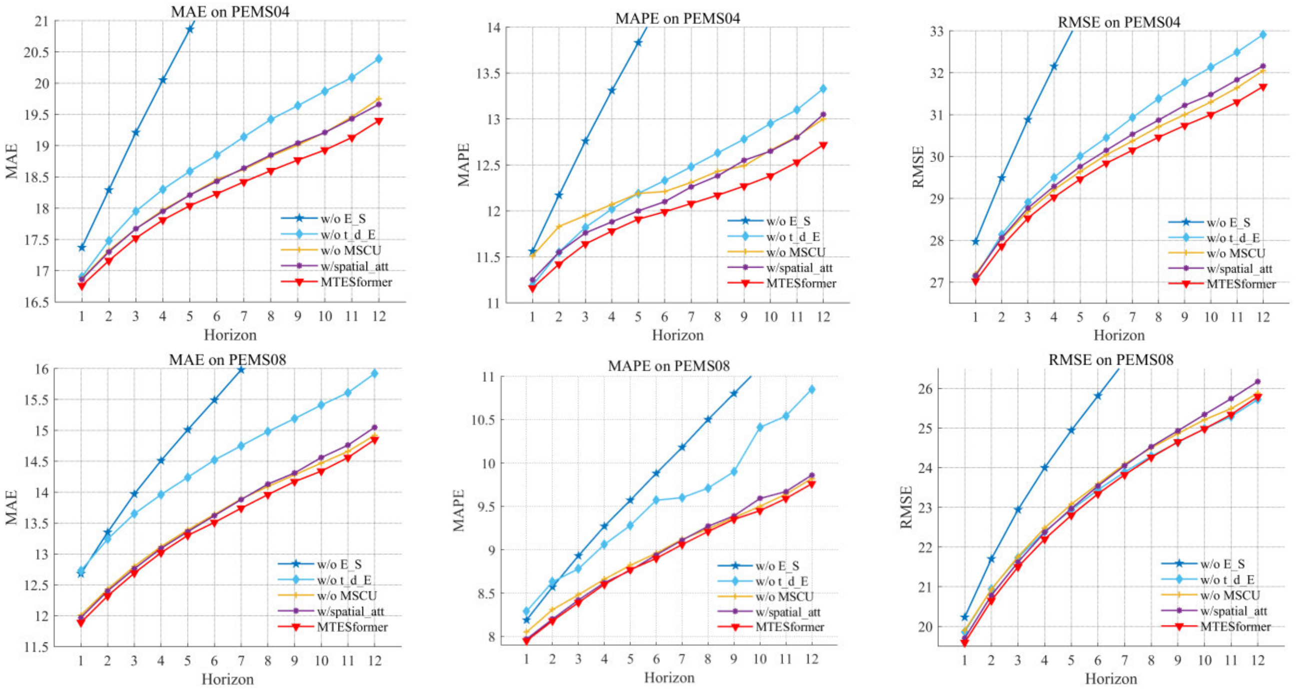


FIGURE 6. Ablation experiment of module effectiveness on PEMS04 and PEMS08.

TABLE 5. Performance on PEMS04 and PEMS08 datasets.

Datasets	PEMS04			PEMS08		
Metric	MAE	RMSE	MAPE	MAE	RMSE	MAPE
w/o t_d_E	18.86	30.51	12.36%	14.50	23.47	9.36%
w/o E_S	21.99	35.20	14.67%	15.66	26.12	10.05%
w/o MSCU	18.45	30.03	12.29%	13.64	23.55	8.97%
w/spatial_att	18.43	30.14	12.19%	13.67	23.51	8.97%
MTESformer	**18.23**	**29.78**	**12.00%**	**13.53**	**23.32**	**8.93%**

(1) w/o t_d_E: After removing the time of day and day of week Embedding, the MAE rises by 0.63 and 0.97 on the PEMS04 and PEMS08 datasets, which suggests that our periodical feature embedding of the flow is reasonable. It can capture daily and weekly traffic patterns and can help the model to better capture temporal dependencies.

(2) w/o E_S: The removal of this module results in a significant decrease in model performance. It not only illustrates the importance of capturing spatial dependencies for model prediction, but also proves that the Enhance Spatial Transformer module we developed does play an important role in model prediction.

(3) w/o MSCU: After removing the MSCU, all three metrics increase on the PEMS04 and PEMS08 datasets, with the MAE increasing by 0.22 and 0.11, respectively. illustrating

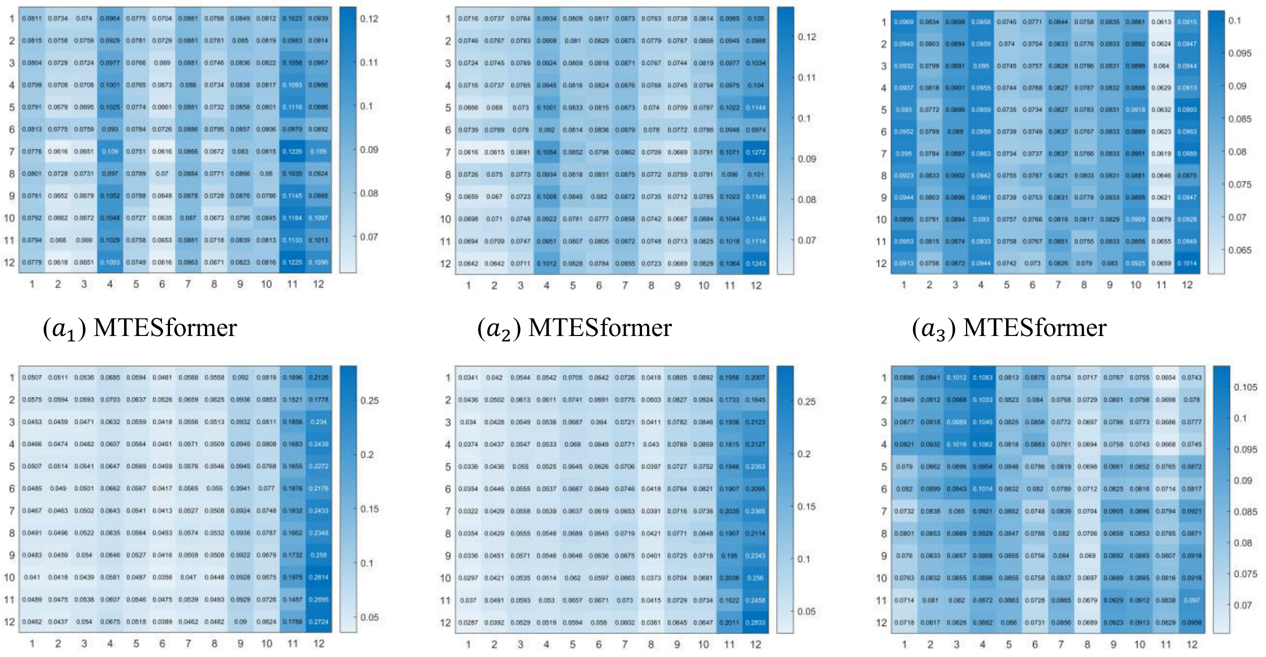
that the MSCU module can discover temporal feature patterns at different scales, further improving the model accuracy based on temporal self-attention.

(4) w/spatial_att: After using the ordinary spatial self-attention module instead of Enhance Spatial Transformer, the prediction accuracy of the model decreases on both datasets. This indicates that Enhance Spatial Transformer we developed is significantly better than the ordinary spatial self-attention module, and that Enhance Spatial Transformer simultaneous focus on long-distance spatial dependencies and neighborhood nodes effectively improves the model accuracy. The 4 sets of comparisons above demonstrate the effectiveness of our model components. These well-designed components are the reason why our model achieves an advantage.

C. VISUALIZATION

1) VISUALIZATION OF TEMPORAL DEPENDENCY

In order to better explore the effectiveness of multi-scale convolution unit (MSCU) in capturing the dynamic time dependence, we conduct experiments on the test set of PEMS04 dataset. As shown in Fig.7, a_1, a_2, a_3 are the heat maps plotted by the temporal self-attention coefficient with the addition of the MSCU, and b_1, b_2, b_3 are the heat maps plotted by the temporal self-attention coefficient with the deletion of the MSCU, a_1 and b_1 are plotted for the same test set of data, a_2 and b_2 are plotted for the same test set of data, a_3 and b_3 are plotted for the same set of test data. It can be seen that in b_1, attention is focused on time points 11 and 12, while with the addition of the MSCU, as shown in a_1, additional attention is paid to time points 4 and 7, on top of the attention being retained at time points 11 and 12. And a_2, b_2 and a_3, b_3 are similar. We believe that the multi-scale convolution unit (MSCU), by discovering temporal pattern features at different scales and joining the temporal self-attention module, can mine more hidden temporal dependencies, better



(a_1) MTESformer (a_2) MTESformer (a_3) MTESformer

(b_1) w/o MSCU (b_2) w/o MSCU (b_3) w/o MSCU

FIGURE 7. a_1, a_2, a_3 **Heatmap of the temporal dependence matrix derived from the temporal self-attention scores of MTESformer on the test set data.** b_1, b_2, b_3 **Heatmap of the temporal self-attention scores derived from the w/o MSCU on the same test set data.**

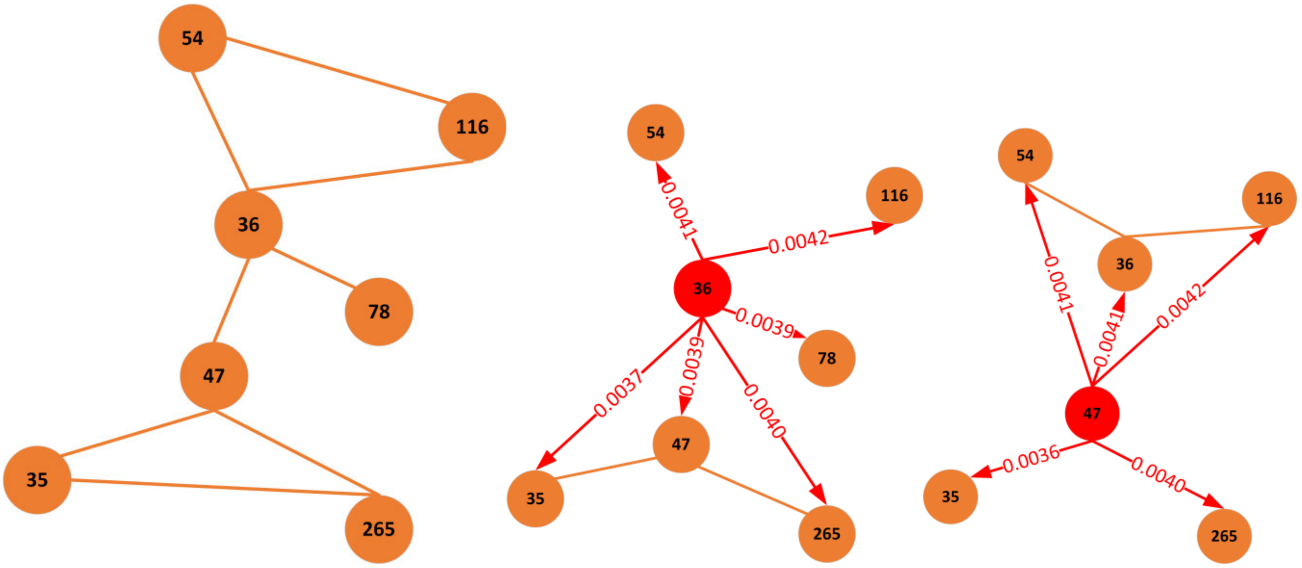
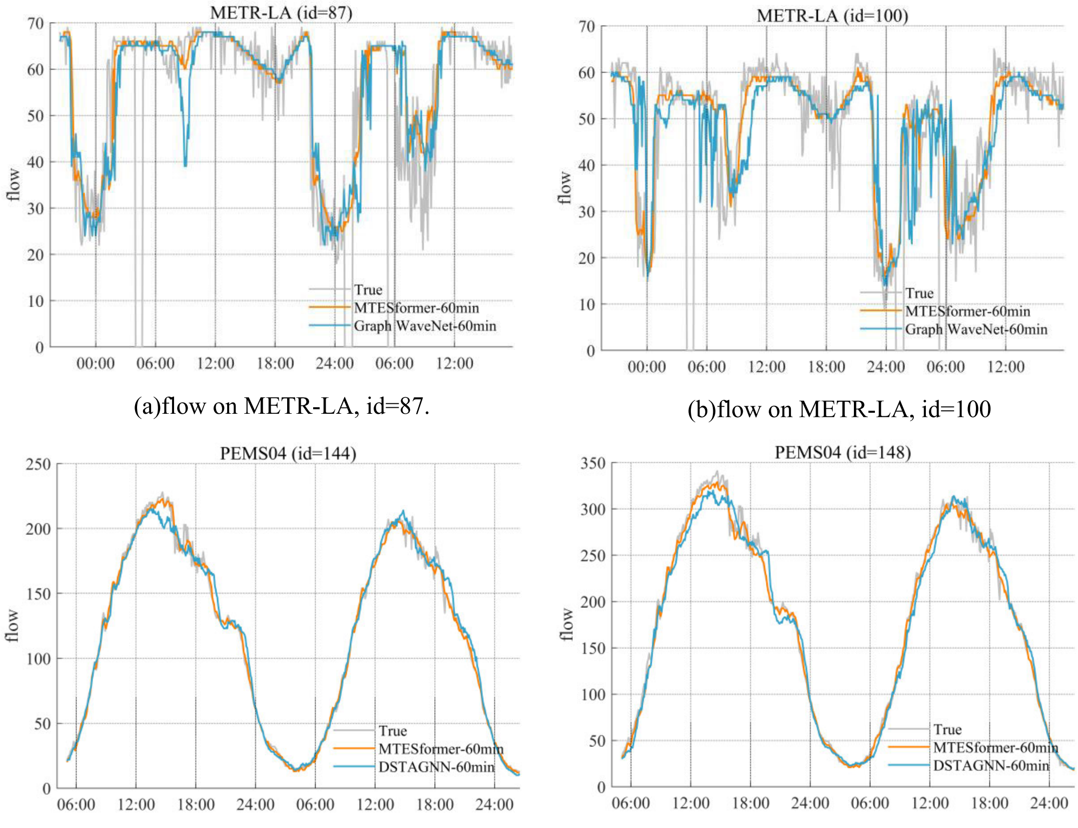


FIGURE 8. (a) Shows the road network connectivity in PEMS04 with number 36 and 47 as the center nodes. (b) and (c) are plots of the spatial attention coefficients of nodes numbered 36 and 47 to the surrounding nodes, respectively, where the values on the red arrows are derived from the spatial self-attention scores in the enhance spatial Transformer in the third spatial-temporal transformer layer.

identify different traffic flow patterns, and help the model extract dynamic temporal dependencies.

2) VISUALIZATION OF SPATIO DEPENDENCY

To further explore the role played by Enhance Spatial Transformer in the model, we conduct experiments on the test set of PEMS04 dataset. As shown in Fig.8, (a) is the schematic diagram of some node adjacencies in the PEMS04 dataset, (b) and (c) are the schematic diagrams of adjacencies for the nodes numbered 36 and 47 as the center node, respectively, and the numbers on the red arrows are the attentional scores of the center node to other nodes. In (b) and (c), it can be seen that the nodes numbered 36 and 47 all have generally high attention coefficients to their closer neighboring nodes,



(a)flow on METR-LA, id=87.

(b)flow on METR-LA, id=100

(c)flow on PMES04, id=144.

(d)flow on PMES04, id=148.

FIGURE 9. Comparison of the prediction results of MTESformer with graph WaveNet is shown on the METR-LA and PMES04 datasets.

indicating that they pay more attention to their surrounding nodes because the traffic changes of the surrounding nodes are more influential to them. The other nodes which also have high attention scores (e.g., node number 36 has attention scores of 0.0042, 0.0040, 0.0041 for nodes such as 11, 30, 44, etc., respectively), we believe that these nodes are nodes which are farther away from node number 36 but have similar functions.

3) VISUALIZATION OF FLOW PREDICTION

In order to better demonstrate the advantages of MTESformer in long-term prediction, we performed visualizations on the PEMS04 and METR-LA datasets, respectively. As shown in Fig.9. We selected two days of complete traffic data on the test set for prediction. We selected Graph WaveNet and DSTAGNN models as a comparison and presented the 60min prediction results. On the METR-LA dataset, MTESformer predicts significantly better than Graph WaveNet during the peak periods of traffic as well as during periods of sudden changes in traffic. Similarly, on the PEMS04 dataset, MTESformer's predictions are closer to the true values during peak periods. From the prediction results, it can be seen that in the long-term prediction of 60 min, MTESformer is more accurate in predicting the peak period as well as the period of sudden change in flow, which indicates that MTESformer can accurately capture the spatial-temporal dependence of the complex time period, and it has an advantage for long-term prediction as well as the prediction of the complex time period.

VI. CONCLUSION AND PROSPECT

In this work, we propose Multi-scale Temporal and Enhance Spatial Transformer (MTESformer) to predict traffic flow conditions on traffic road networks. Specifically, we make reasonable use of the periodic characteristics of the traffic flow; we develop a Multi-scale Temporal Transformer, which effectively focuses on the temporal correlations in time series data, it combines the multi-head self-attention mechanism with multi-scale convolution units (MSCU) to better identify different traffic flow patterns and capture long-term dependencies; we have also developed an Enhance Spatial Transformer, which incorporates node adjacencies into the attention mechanism to enhance attention to nearby nodes and capture long distance spatial dependencies more effectively. We conducted extensive experiments on four real-world datasets, and our model exceeds most of the baseline levels and has a great advantage in long-term prediction, proving the superiority of our proposed MTESformer model, in addition to visualizing the learned attention. However, our model does not take into account the effects of external factors such as unexpected events, special weather conditions, road conditions, and regional attributes. In the future, we would like to incorporate more effective external factors into the model and study how to use these external factors correctly to make the prediction more accurate. Meanwhile, it is hoped that MTESformer can be applied to other spatial-temporal predictions, and pre-training techniques in the field of traffic prediction will also be explored to solve the challenges of missing data and to reduce the task burden.

REFERENCES

[1] D. A. Tedjopurnomo, Z. Bao, B. Zheng, F. M. Choudhury, and A. K. Qin, ''A survey on modern deep neural network for traffic prediction: Trends, methods and challenges,'' *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 4, pp. 1544–1561, Apr. 2022, doi: 10.1109/TKDE.2020.3001195.

[2] X. Yin, G. Wu, J. Wei, Y. Shen, H. Qi, and B. Yin, ''Deep learning on traffic prediction: Methods, analysis, and future directions,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 4927–4943, Jun. 2022, doi: 10.1109/TITS.2021.3054840.

[3] H. Yao, X. Tang, H. Wei, G. Zheng, and Z. Li, ''Revisiting spatial–temporal similarity: A deep learning framework for traffic prediction,'' in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, no. 1, pp. 5668–5675, doi: 10.1609/aaai.v33i01.33015668.

[4] J. Zhang, Y. Zheng, and D. Qi, ''Deep spatio-temporal residual networks for citywide crowd flows prediction,'' in *Proc. AAAI Conf. Artif. Intell.*, vol. 31, no. 1, Feb. 2017, pp. 1–7, doi: 10.1609/aaai.v31i1.10735.

[5] Y. Chen, J. Guo, H. Xu, J. Huang, and L. Su, ''Improved long short-term memory-based periodic traffic volume prediction method,'' *IEEE Access*, vol. 11, pp. 103502–103510, 2023, doi: 10.1109/ACCESS.2023.3305398.

[6] H. Yao, F. Wu, J. Ke, X. Tang, Y. Jia, S. Lu, P. Gong, J. Ye, and Z. Li, ''Deep multi-view spatial–temporal network for taxi demand prediction,'' in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2018, vol. 2018, no. 1, pp. 1–8, doi: 10.1609/aaai.v32i1.11836.

[7] Y. Seo, M. Defferrard, P. Vandergheynst, and X. Bresson, ''Structured sequence modeling with graph convolutional recurrent networks,'' in *Proc. Int. Conf. Neural Inf. Process.* Springer, 2018, pp. 362–373.

[8] Y. Li, R. Yu, C. Shahabi, and Y. Liu, ''Diffusion convolutional recurrent neural network: Data-driven traffic forecasting,'' 2017, *arXiv:1707.01926*.

[9] Z. Pan, Y. Liang, W. Wang, Y. Yu, Y. Zheng, and J. Zhang, ''Urban traffic prediction from spatio-temporal data using deep meta learning,'' in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Min.*, Jul. 2019, pp. 1720–1730, doi: 10.1145/3292500.3330884.

[10] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, ''Graph WaveNet for deep spatial–temporal graph modeling,'' 2019, *arXiv:1906.00121*.

[11] Z. Fang, Q. Long, G. Song, and K. Xie, ''Spatial–temporal graph ODE networks for traffic flow forecasting,'' in *Proc. 27th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2021, pp. 364–373, doi: 10.1145/3447548.3467430.

[12] J. Choi, H. Choi, J. Hwang, and N. Park, ''Graph neural controlled differential equations for traffic forecasting,'' in *Proc. AAAI Conf. Artif. Intell.*, vol. 36, no. 6, Jun. 2022, pp. 6367–6374, doi: 10.1609/aaai.v36i6.20587.

[13] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, ''WaveNet: A generative model for raw audio,'' 2016, *arXiv:1609.03499*.

[14] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, ''Informer: Beyond efficient transformer for long sequence time-series forecasting,'' in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 12, 2021, pp. 11106–11115.

[15] B. Pan, U. Demiryurek, and C. Shahabi, ''Utilizing real-world transportation data for accurate traffic prediction,'' in *Proc. IEEE 12th Int. Conf. Data Mining*, Dec. 2012, pp. 595–604.

[16] Y. Sun, G. Zhang, and H. Yin, ''Passenger flow prediction of subway transfer stations based on nonparametric regression model,'' *Discrete Dyn. Nature Soc.*, vol. 2014, pp. 1–8, Apr. 2014, doi: 10.1155/2014/397154.

[17] C.-H. Wu, J.-M. Ho, and D. T. Lee, ''Travel-time prediction with support vector regression,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 4, pp. 276–281, Dec. 2004, doi: 10.1109/TITS.2004.837813.

[18] E. Zivot and J. Wang, ''Vector autoregressive models for multivariate time series,'' in *Modeling Financial Time Series With S-PLUS*. Cham, Switzerland: Springer, 2006, pp. 385–429.

[19] Z. Zheng and D. Su, ''Short-term traffic volume forecasting: A *k*-nearest neighbor approach enhanced by constrained linearly sewing principle component algorithm,'' *Transp. Res. Part C: Emerg. Technol.*, vol. 43, pp. 143–157, Jun. 2014, doi: 10.1016/j.trc.2014.02.009.

[20] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, ''Convolutional sequence to sequence learning,'' 2017, *arXiv:1705.03122*.

[21] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang, ''Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction,'' *Sensors*, vol. 17, no. 4, p. 818, Apr. 2017, doi: 10.3390/s17040818.

[22] G. Shen, C. Chen, Q. Pan, S. Shen, and Z. Liu, ''Research on traffic speed prediction by temporal clustering analysis and convolutional neural network with deformable kernels,'' *IEEE Access*, vol. 6, pp. 51756–51765, 2018, doi: 10.1109/ACCESS.2018.2868735.

[23] A. Khan, M. M. Fouda, D.-T. Do, A. Almaleh, and A. U. Rahman, ''Short-term traffic prediction using deep learning long short-term memory: Taxonomy, applications, challenges, and future trends,'' *IEEE Access*, vol. 11, pp. 94371–94391, 2023, doi: 10.1109/ACCESS.2023.3309601.

[24] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, ''Empirical evaluation of gated recurrent neural networks on sequence modeling,'' 2014, *arXiv:1412.3555*.

[25] S. Hochreiter and J. Schmidhuber, ''Long short-term memory,'' *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[26] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, ''Long short-term memory neural network for traffic speed prediction using remote microwave sensor data,'' *Transp. Res. Part C, Emerg. Technol.*, vol. 54, pp. 187–197, May 2015, doi: 10.1016/j.trc.2015.03.014.

[27] K. Li, W. Bai, S. Huang, G. Tan, T. Zhou, and K. Li, ''Lag-related noise shrinkage stacked LSTM network for short-term traffic flow forecasting,'' *IET Intell. Transp. Syst.*, vol. 18, no. 2, pp. 244–257, Nov. 2023, doi: 10.1049/itr2.12448.

[28] S. Bai, J. Z. Kolter, and V. Koltun, ''An empirical evaluation of generic convolutional and recurrent networks for sequence modeling,'' 2018, *arXiv:1803.01271*.

[29] J. Bi, J. Zhang, H. Yuan, and J. Qiao, ''Integrated spatio-temporal prediction for water quality with graph attention network and WaveNet,'' in *Proc. IEEE Int. Conf. Syst.*, Nov. 2022, pp. 2551–2556, doi: 10.1109/SMC53654.2022.9945240.

[30] D. Yang and L. Lv, ''A graph deep learning-based fast traffic flow prediction method in urban road networks,'' *IEEE Access*, vol. 11, pp. 93754–93763, 2023, doi: 10.1109/ACCESS.2023.3308238.

[31] W. Chen, L. Chen, Y. Xie, W. Cao, and X. Feng, ''Multi-range attentive bicomponent graph convolutional network for traffic forecasting,'' in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 4, 2020, pp. 3529–3536.

[32] D. Liu, J. Wang, S. Shang, and P. Han, ''MSDR: Multi-step dependency relation networks for spatial–temporal forecasting,'' in *Proc. 28th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2022, pp. 1042–1050, doi: 10.1145/3534678.3539397.

[33] C. Zheng, X. Fan, C. Wang, and J. Qi, ''GMAN: A graph multi-attention network for traffic prediction,'' in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 1, Apr. 2020, pp. 1234–1241, doi: 10.1609/aaai.v34i01.5477.

[34] X. Ye, S. Fang, F. Sun, C. Zhang, and S. Xiang, ''Meta graph transformer: A novel framework for spatial–temporal traffic prediction,'' *Neurocomputing*, vol. 491, pp. 544–563, Jun. 2022, doi: 10.1016/j.neucom.2021.12.033.

[35] J. Zhu, X. Han, H. Deng, C. Tao, L. Zhao, P. Wang, T. Lin, and H. Li, ''KST-GCN: A knowledge-driven spatial–temporal graph convolutional network for traffic forecasting,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 15055–15065, Sep. 2022, doi: 10.1109/TITS.2021.3136287.

[36] A. H. A. Hussain, M. A. Taher, O. A. Mahmood, Y. I. Hammadi, R. Alkanhel, A. Muthanna, and A. Koucheryavy, ''Urban traffic flow estimation system based on gated recurrent unit deep learning methodology for Internet of Vehicles,'' *IEEE Access*, vol. 11, pp. 58516–58531, 2023, doi: 10.1109/ACCESS.2023.3270395.

[37] F. Li, J. Feng, H. Yan, G. Jin, F. Yang, F. Sun, D. Jin, and Y. Li, ''Dynamic graph convolutional recurrent network for traffic prediction: Benchmark and solution,'' *ACM Trans. Knowl. Discovery Data*, vol. 17, no. 1, pp. 1–21, Feb. 2023, doi: 10.1145/3532611.

[38] Q. Zhang, L. Zhou, Y. Su, H. Xia, and B. Xu, ''Gated recurrent unit embedded with dual spatial convolution for long-term traffic flow prediction,'' *ISPRS Int. J. Geo-Inf.*, vol. 12, no. 9, p. 366, Sep. 2023, doi: 10.3390/ijgi12090366.

[39] B. Hussain, M. Khalil Afzal, S. Anjum, I. Rao, and B.-S. Kim, "A novel graph convolutional gated recurrent unit framework for network-based traffic prediction," *IEEE Access*, vol. 11, pp. 130102–130118, 2023, doi: 10.1109/ACCESS.2023.3333938.

[40] K. Cao, Y. Liu, L. Duan, S. Xu, and H. Jung, "Research on regional traffic flow prediction based on MGCN-WOALSTM," *IEEE Access*, vol. 11, pp. 126436–126446, 2023, doi: 10.1109/ACCESS.2023.3330909.

[41] Q. Zhang, M. Tan, C. Li, H. Xia, W. Chang, and M. Li, "Spatio-temporal residual graph convolutional network for short-term traffic flow prediction," *IEEE Access*, vol. 11, pp. 84187–84199, 2023, doi: 10.1109/ACCESS.2023.3300232.

[42] W. Xiao and X. Wang, "Spatial–temporal dynamic graph convolutional neural network for traffic prediction," *IEEE Access*, vol. 11, pp. 97920–97929, 2023, doi: 10.1109/ACCESS.2023.3312534.

[43] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," 2015, *arXiv:1511.07122*.

[44] X. Kong, W. Xing, X. Wei, P. Bao, J. Zhang, and W. Lu, "STGAT: Spatial–temporal graph attention networks for traffic flow forecasting," *IEEE Access*, vol. 8, pp. 134363–134372, 2020, doi: 10.1109/ACCESS.2020.3011186.

[45] P. Velič ković, G. Cucurull, A. Casanova, A. Romero, P. Lió, and Y. Bengio, "Graph attention networks," 2017, *arXiv:1710.10903*.

[46] Y. Shin and Y. Yoon, "PGCN: Progressive graph convolutional networks for spatial–temporal traffic forecasting," 2022, *arXiv:2202.08982*.

[47] A. Prabowo, W. Shao, H. Xue, P. Koniusz, and F. D. Salim, "Because every sensor is unique, so is every pair: Handling dynamicity in traffic forecasting," in *Proc. 8th ACM/IEEE Conf. Internet Things Des. Implement.*, May 2023, pp. 93–104, doi: 10.1145/3576842.3582362.

[48] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017, *arXiv:1706.03762*.

[49] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[50] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[51] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," 2021, *arXiv:2103.14030*.

[52] N. Wang and X. Zhao, "Enformer: Encoder-based sparse periodic self-attention time-series forecasting," *IEEE Access*, vol. 11, pp. 112004–112014, 2023, doi: 10.1109/ACCESS.2023.3322957.

[53] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial–temporal graph convolutional networks for traffic flow forecasting," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, no. 1, pp. 922–929.

[54] Z. Zhao, L. Chao, X. Zhang, N. Xie, and Q. Zeng, "MCAGCN: Multi-component attention graph convolutional neural network for road travel time prediction," *IET Intell. Transp. Syst.*, vol. 18, no. 1, pp. 139–153, Nov. 2023, doi: 10.1049/itr2.12440.

[55] M. Xu, W. Dai, C. Liu, X. Gao, W. Lin, G.-J. Qi, and H. Xiong, "Spatial–temporal transformer networks for traffic flow forecasting," 2020, *arXiv:2001.02908*.

[56] J. Zhang, J. Jin, J. Tang, and Z. Qu, "FPTN: Fast pure transformer network for traffic flow forecasting," in *Proc. Int. Conf. Artif. Neural Netw.*, Sep. 2023, pp. 382–393, doi: 10.1007/978-3-031-44223-0_31.

[57] A. Feng and L. Tassiulas, "Adaptive graph spatial–temporal transformer network for traffic flow forecasting," 2022, *arXiv:2207.05064*.

[58] S. Lan, Y. Ma, W. Huang, W. Wang, H. Yang, and P. Li, "DSTAGNN: Dynamic spatial–temporal aware graph neural network for traffic flow forecasting," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2022, pp. 11906–11917.

[59] X. Luo, C. Zhu, D. Zhang, and Q. Li, "Dynamic graph convolutional network with attention fusion for traffic flow prediction," 2023, *arXiv:2302.12598*.

[60] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," 2014, *arXiv:1409.3215*.

[61] L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," 2020, *arXiv:2007.02842*.

[62] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," 2017, *arXiv:1709.04875*.

[63] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, and C. Zhang, "Connecting the dots: Multivariate time series forecasting with graph neural networks," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Min.*, Aug. 2020, pp. 753–763, doi: 10.1145/3394486.3403118.

[64] J. Jiang, C. Han, W. X. Zhao, and J. Wang, "PDFormer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction," in *Proc. AAAI Conf. Artif. Intell.*, 2023, pp. 4365–4373, doi: 10.1609/aaai.v37i4.25556.

XINHUA DONG received the Ph.D. degree from the School of Computer Science and Technology, Huazhong University of Science and Technology, China, in 2016. He is currently a Lecturer (Master Tutor) with the School of Computer Science, Hubei University of Technology. His research interests include machine learning, big data management, cloud computing, distributed system security, and intelligent information processing. He is a member of China Computer Federation.

WANBO ZHAO is currently pursuing the B.S. degree with the School of Computer Science, Hubei University of Technology. His current research interests include deep learning and traffic flow forecasting in intelligent transportation systems.

HONGMU HAN received the Ph.D. degree from the School of Computer Science and Technology, Huazhong University of Science and Technology, China, in 2018. He is currently a Lecturer (Master Tutor) with the School of Computer Science, Hubei University of Technology. His main research interests include blockchain, information security, and mobile security.

ZHANYI ZHU is currently pursuing the M.S. degree with the School of Computer Science, Hubei University of Technology. Her current research interests include machine learning and traffic flow forecasting in intelligent transportation systems.

HUI ZHANG is currently pursuing the M.S. degree with the School of Computer Science, Hubei University of Technology. His current research interests include deep learning and intelligent information processing.