HMIAN: a Hierarchical Mapping and Interactive Attention Data Fusion Network for Traffic Forecasting

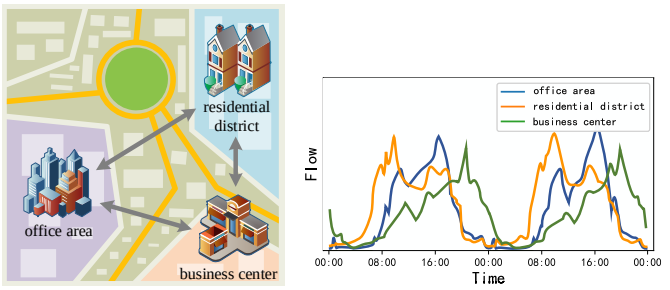
Jingru Sun, *Member, IEEE,* Mu Peng, Hongbo Jiang, *Senior Member, IEEE,* Qinghui Hong, and Yichuang Sun, *Senior Member, IEEE*

***Abstract*—With the development of intelligent transportation system (ITS), the vital technology of ITS, short-term traffic forecasting, gains increasing attention. However, the existing prediction models ignore the impact of urban functional zones on traffic data, resulting in inaccurate extractions of dynamic spatial relationships from network. Furthermore, how to calculate the influence of external factors such as weather and holidays on traffic is an unsolved problem. This paper proposes a spatio-temporal hierarchical mapping and interactive attention network (HMIAN), which extracts the spatial features from traffic network by constructing functional zones, and designs an effective external factors fusion method. HMIAN uses the hierarchical mapping structure to aggregate the roads into functional zones, calculate the interaction between functional zones and feed this information back to the spatial features. And the interactive attention mechanism is utilized to fuse the traffic data with external factors effectively, and extracts temporal features. In addition, some experiments were carried out on three real traffic data sets. First, experiment results show that the proposed model better prediction performance compared with other existing approaches in more complex traffic network. Second, the longitudinal comparison experiment verifies that the hierarchical mapping structure is effective in extracting spatial features in complex road network. Finally, the influence of different external factors and fusion methods on traffic prediction are compared, which provides a consult for subsequent research on the influence of external factors.**

***Index Terms*—traffic prediction, deep learning, data fusion.**

I. Introduction

WITH the development of emerging technologies such as Internet of things (IoT) and artificial intelligence (AI), intelligent transportation system (ITS) is considered to be one of the most promising applications in in the transportation field [1], [2], [3]. As an important part of ITS, traffic prediction provides necessary technical support for urban traffic planning, public travel and traffic accident avoidance, which includes traffic flow, average speed, occupancy prediction. Traffic forecasting methods can be roughly divided into two categories, deductive and inductive [4]. Deductive methods are not applicable to the actual complex traffic network, because it is mainly based on the simulation of traffic situations. The mainstream forecasting algorithm are inductive methods in recent years.



(a) Functional zones interact with each other.

(b) Traffic flow in different functional zones.

Fig. 1. Adjacent roads with correlation form functional zones with certain structure, and different zones present different traffic trends due to differences in functional characteristics.

Inductive models can also be classified into model-driven and data-driven methods [5]. model-driven methods predict the future traffic data by analyzing the change trend of the sequence. For example, Autoregressive Integrated Moving Average (ARIMA) [6] model, Kalman filter and its derivatives [7], [8] have been widely used in traffic forecasting [9]. Nevertheless, due to the uncertainty and nonlinearity of traffic information, model-driven methods are difficult to effectively capture the regularity of traffic changes from the data.

In order to deal with nonlinear data, data-driven methods which consist of machine learning method and deep learning method are applied to traffic prediction [10], [11], [12]. Machine learning models, such as Support Vector Machine (SVM) [13], [14], k-Nearest Neighbor algorithm [15] and k-means clustering algorithm [16], shows good performance in predicting short-term traffic data compared to model-driven models. However, the complexity of traffic data of real network has high complexity, hence these traditional shallow machine learning methods are difficult to make more accurate predictions.

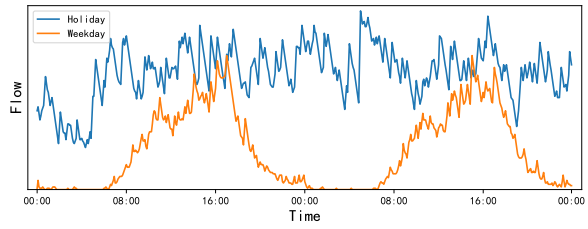
The application of deep learning in traffic prediction improves the efficiency of processing the traffic data with high

This work is supported in part by the National Natural Science Foundation of China under Grant 62171182, in part by the Natural Science Foundation of Hunan Province under Grant 2021JJ30147, in part by the Science and Technology Progress and Innovation Project of Hunan Transportation Department of China under Grant 2018037. (Corresponding author: Qinghui Hong.)

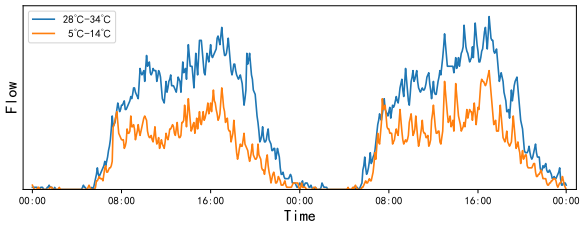
J. Sun, M. Peng and Q. Hong are with the Department of Communication Engineering, College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China (e-mail: jt_sunjr@hnu.edu.cn; pengmu97@hnu.edu.cn; hongqinghui@hnu.edu.cn).

H. Jiang is with the Department of Software Engineering, College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China (e-mail: hongbojiang@hnu.edu.cn).

Y. Sun is with the School of Engineering and Computer Science, University of Hertfordshire, Hatfield AL109AB, U.K. (e-mail: y.sun@herts.ac.uk).



(a) Traffic flow during holidays and weekday.



(b) Traffic flow at different temperatures.

Fig. 2. External factors, such as holidays and weather, have a certain impact on the traffic data.

complexity and can represent traffic features without prior knowledge [17], such as stacked autoencoder (SAE) [17] and deep belief network (DBN) [18]. Furthermore, traffic data as time series, to better extract time correlation from traffic data, recurrent neural network (RNN) derivative algorithms, such as long short-term memory (LSTM) model [19] neural network and gated recurrent unit networks (GRU) [20], [21], have been introduced. Nonetheless, the traffic prediction in a certain road is not only influenced by historical data, but also be influenced by the traffic conditions of surrounding roads, namely, the spatial-temporal features of traffic data [22], [23]. The above algorithms ignore the spatial influence on traffic prediction.

It is necessary to process the structure of traffic network to extract the spatial representation from traffic data. Convolutional Neural Network (CNN) is applied in this field for its local feature extraction capability [24]. For instance, LC-RNN [25] is a hybrid model composed of RNN and CNN, which can capture multidimensional dependencies, and MGSTC [26] can explore multiple spatio-temporal dependencies through multiple gated spatio-temporal CNN branches. Since the traffic network is a non-Euclidean structure, researchers need to preprocess the road network into a matrix like Euclidean structure according to the number of neighbors of all node [27], [28]. In fact, the actual traffic network structure is very complex, and this kind of preprocessing to transform the road network into Euclidean structure brings more difficulties to the prediction.

To capture the spatial features of non-Euclidean traffic network more effectively, researchers use graph convolution network (GCN) to process the road network [29]. GCN converts network data from spatial domain to spectral domain, which efficiently processes topological data to extract spatial features [30]. STGCN [31] and GraphWaveNet [32] model use GCN to capture the comprehensive temporal and spatial correlation of multi-scale traffic network. Besides, random walk algorithm can also be used to calculate the spatial correlation of traffic network. DCRNN [33] model uses bidirectional random walk

TABLE I
THE KEY NOTATIONS

Notations	Description
$G = \langle V, A \rangle$	The graph of traffic road network structure.
V	The set of sensors deployed on the road network.
A	The adjacency matrix of nodes.
$\mathcal{X}, \mathcal{Y}$	Input data and forecast result.
t	Time step.
τ	The number of time steps of data.
c	The number of types of features.
E	Feature embedding of data.
M	Hard mapping matrix.
g	The gate to control the fusion of information.
W, b, w	The trainable parameters.

to model the spatial dependence. Even so, the existing graph neural networks rely too much on the fixed road network structure and ignore the dynamic changes of the connections (weights) between nodes resulting in the decline of prediction accuracy and lack of generality.

The Graph Attention Networks (GAT) model was proposed by Velickovic al et. [34], which can aggregate the weights of adjacent node features by attention mechanism, and the weights of adjacent node features depend entirely on node features and are independent of graph structure [35]. GMAN [36] model was proposed to extract dynamic spatial correlation through graph multi-head attention mechanism. Therefore, in traffic prediction, the universality of the model under different road network structures can be improved.

The urban traffic can be divided into different zones according to different functional features. As shown in Fig. 1, the traffic features of roads within the regions are similar, that is, the structure of road network shows the features of clustering, and each cluster has its own structure (e.g., overpass) and function (e.g., residential district) [37]. In addition, due to the interaction between functional zones, there are dependencies between long-distance roads [38]. But, in the current research on traffic prediction, these problems are not considered in forecasting methods.

From the perspective of data other than traffic flow, existing forecasting methods tend to ignore the impact of external factors, such as holidays and weather [39]. These factors have different degrees of influence on different areas. As shown in Fig. 2(a), during the holidays, the traffic flow shows a surge, and the traffic trend no longer has obvious regularity. Fig. 2(b) shows that weather factors also have much influence on traffic flow.

To solve the above problems, this paper proposes a hierarchical mapping and interactive attention network (HMIAN) data fusion prediction model. In terms of extracting spatial features, the model uses hierarchical mapping structure takes to capture the relevance of functional zones and long-distance dependence on traffic network. In terms of temporal features, the model uses the interactive attention mechanism to calculate the influence of external factors on traffic data. The main contributions of this paper can be summarized as follows:

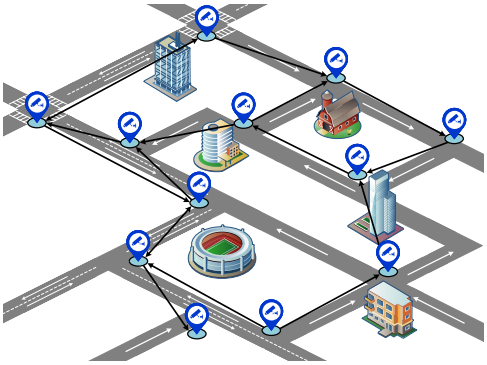


Fig. 3. The transformation between real road network structure and adjacency graph. The white arrow represents the direction of the road, and the monitor icon represents the location of the sensor. Each sensor can be regarded as a node, and the black arrow indicates whether it can be directly reached between nodes.

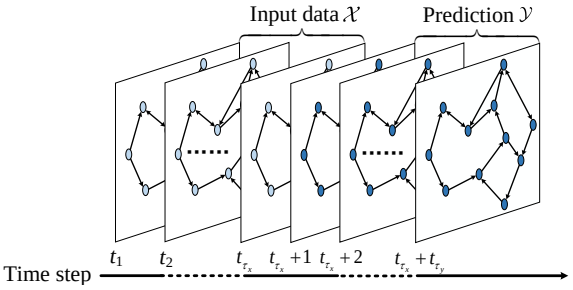


Fig. 4. Traffic data represented in time dimension. Each piece of graph denotes the traffic information of the network at a certain time step. The prediction task is using τ_x time steps historical data ($\mathcal{X}$) to predict next τ_y time steps traffic data ($\mathcal{Y}$).

1) Hierarchical mapping network is proposed to aggregate the road network structure to construct functional zones. By calculating the interaction between functional zones, distant dependencies between roads are captured and feed them back to spatial features through the top-down updating mechanism.
2) HMIAN model adopts interactive attention mechanism and intra-attention mechanism, abstracts traffic data and external factor data into embedding sequences, which improves the degree of fusion and the accuracy of prediction.
3) Experiment results shows that the HMIAN model has better performance in the complex traffic network compared with other baseline models. Moreover, the longitudinal comparison experiments verify that the interaction of functional zones can affect the traffic prediction, and compare the influence of different data fusion methods and different external factors on the predicted results. the results demonstrate that the hierarchical mapping network and the interactive attention can improve the prediction accuracy.

The follow-up content of the article is organized as below. Section II gives some definitions that need to be understood before introducing the model. The model proposed in this paper is introduced in detail in section III. Section IV mainly introduces the experiments to verify the effectiveness of the model and the analysis of the results. Finally, section V summarizes the full paper.

II. Preliminarise

This section introduces some notations and definitions that are used in the model, and defines the task of the model. Some important notations are recorded in Table I.

Definition 1. Road network structure. The structure of the traffic network can be regarded as a weighted directed graph $G = \langle V, A \rangle$, where G represents the road network (RN), and V represents the set of nodes. In this paper, all corresponding traffic sensors in the data set can be equivalent to nodes in the directed graph, which are denoted as $v_i \in V$, $i \in (1, 2, \ldots, n)$, where n is total number of sensors. Another notation A represents the adjacency matrix of nodes in the graph, whose non-zero elements represents the direct distance between nodes. Namely if $A[v_i, v_j] > 0$, $i, j \in (1, 2, \ldots, n)$, vehicles can arrive node v_j directly from node v_i without passing through other nodes. In Fig. 3, for example, the monitor icons represent sensors deployed on the road network, and the white arrow indicates the direction of the lane. According to the above rules, the directed graph of road network can be obtained.

Definition 2. Hierarchical mapping network. This method is used in spatial feature extraction module of the model in this paper. The network is divided into three floors. The first floor represents the original road network (RN) structure, namely G mentioned above. The second floor is a network composed of virtual nodes called as "structural area (SA) nodes" (denoted as V_{SA}), which is obtained through clustering the nodes of the first floor. These structural areas have dense sensor nodes, which play an important role in traffic, such as vertical intersections and viaducts. The virtual nodes of the last floor is called as "functional zone (FZ) nodes" (denoted as V_{FZ}), which is constructed by making further abstraction on the second floor. It can be understood as zones with certain functions, such as business center and residential district. This network structure also solves the problem of dependence between remote nodes, because this kind of abstract information is shared in the nodes of the third floor, and then transmits the information to the underlying nodes through the update mechanism. Details are described in the following sections.

Definition 3. Traffic data and external factors. There are two types of traffic data in this paper. First is the traffic data collected by sensors in the road network, denoted as $\mathcal{X} = \left(X_{t_1}, X_{t_2}, \ldots, X_{t_{\tau_x}}\right) \in \mathbb{R}^{\tau_x \times n \times c}$, where τ_x represents the time steps of historical data entered, n is the total number of sensors and c is described as the number of kinds of features (e.g., speed, traffic flow, occupancy). The prediction results in the next τ_y time steps are denoted as $\mathcal{Y} = \left(Y_{t_{\tau_x}+1}, Y_{t_{\tau_x}+2}, \ldots, Y_{t_{\tau_x}+t_{\tau_y}}\right) \in \mathbb{R}^{\tau_y \times n \times c}$. Another type of data is external factors, including date, time, weather, holiday, etc., which have impacts on traffic. Its data format is described as $\mathcal{X}_{EF} = \left(X^{EF}_{t_{\tau_x}+1}, X^{EF}_{t_{\tau_x}+2}, \ldots, X^{EF}_{t_{\tau_x}+t_{\tau_y}}\right) \in \mathbb{R}^{\tau_y \times c_{EF}}$, where c_{EF} is the numbers of types of external factors. To calculate the impact of external factors on the prediction

results, the time steps of the external factors is consistent with the prediction result.

Prediction task. Given the historical traffic data in continuous τ_x time steps, the external factors in next τ_y time steps and the traffic network structure G, predict the traffic information in future τ_y time steps (such as flow, speed, occupancy rate, etc.).

III. Hierarchical Mapping and Interactive Attention Network

The network model is mainly composed of two-layer HMIA modules, as shown in Fig. 5, and each HMIA module includes a hierarchical mapping network spatial module, a interactive attention temporal module and a gated fusion module, which are used to fuse external factors with traffic date and extract temporal and spatial features. The extracted features form the final result processed by the gating fusion module. The construction and the training method of these modules are described in detail next.

A. Hierarchical Mapping Spatial Module

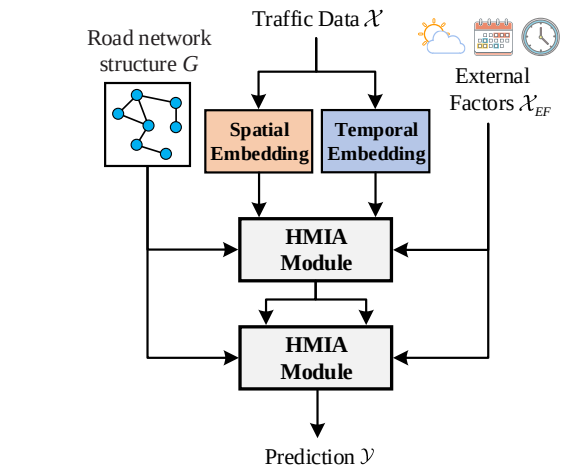
This module uses hierarchical mapping algorithms to abstract the original road network structure, and constructs the structural area floor and functional zone floor, as shown in Fig. 6. This module is mainly composed of two parts. One is the cluster mechanism which is responsible for aggregating node information and constructing the virtual nodes to form the floor of structure area and functional zone mentioned above. Another part of this module is to update the information of each floor of new nodes from top to bottom. This section will explain how each floor is constructed on the basis of the previous floor, and how to update the node information in this multi-level module.

1) Construct Structural Areas: In this paper, structural areas are considered to play an important role in traffic for connection, such as crossroads and viaducts. Assume that each sensor node is located only belongs to the only structure area, and different areas have different importance.

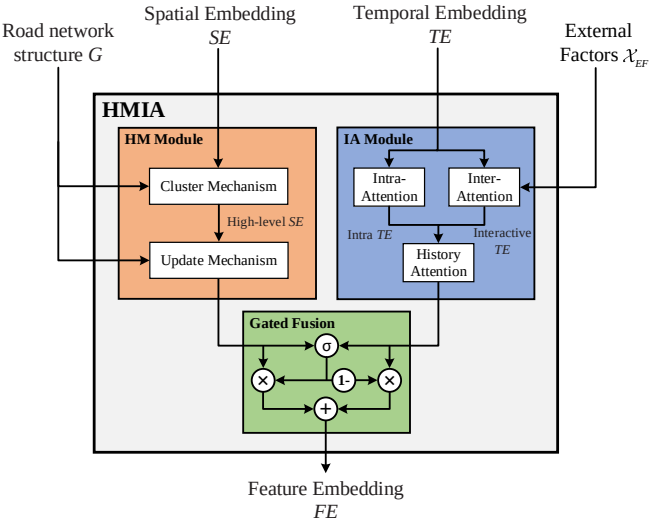
The first operation constructs the mapping matrix between the original road network and the structure area floor by aggregating the node features (e.g., average speed, flow detected by sensors) with the road network structure. Node2vec [40] can be used to cluster the nodes on the road network. This algorithm is only based on the fixed original road network structure, which is used to find the central nodes of clusters and the neighbors closely associated with each node and build a hard mapping matrix from the original network to structural area. The hard mapping matrix is denoted as $M_1 \in \mathbb{R}^{n \times n_{SA}}$, where each element can be represented as follows

$$M_1[v, v_{SA}] = \begin{cases} 1 & v \in v_{SA}, \\ 0 & else, \end{cases} \tag{1}$$

where v and v_{SA} represent the nodes of the original road network and structural area floor respectively. Based on the previous assumption, each sensor node belongs to only one area, namely, only one element whose value is 1 in each row of the matrix.



(a) The Structure of HMIAN.



(b) Internal structure of HMIA module.

Fig. 5. Display of HMIAN model structure and main modules of the model. (a) HMIAN is mainly composed of spatial-temporal embedding module and double-layer HMIA spatial-temporal module. (b) HMIA module consists of hierarchical mapping spatial module, interactive attention temporal module and gated fusion module.

In addition, from the perspective of features of nodes, the input data is first extracted by the fully connected network (FCN) and transformed into spatial embedding (SE). Then this module adopts the Graph Attention Network (GAT) [34] calculate the information interaction embedding (IE) between nodes and surrounding nodes as follows:

$$IE_1 = \mathrm{GAT}\left(SE_{t_i}, A\right), \tag{2}$$

where $SE_{t_i} \in \mathbb{R}^{n \times d_1}$ $(i \in \{1, 2, \ldots, \tau_x\})$ is the high dimensional features matrix of all sensor nodes in time step t_i, d_1 is a constant, A is the adjacency matrix for the road network, and $IE_1 \in \mathbb{R}^{n \times n_{SA}}$ is the learned output through GAT. Here, the dimension of embedding of each node is set to n_{SA} in IE_1 to associate the nodes with the structure areas. The elements of each column vector in IE_1 can be understood as the importance of different nodes in the structural area. Multiply IE_1 and hard mapping matrix M_1, then obtain the

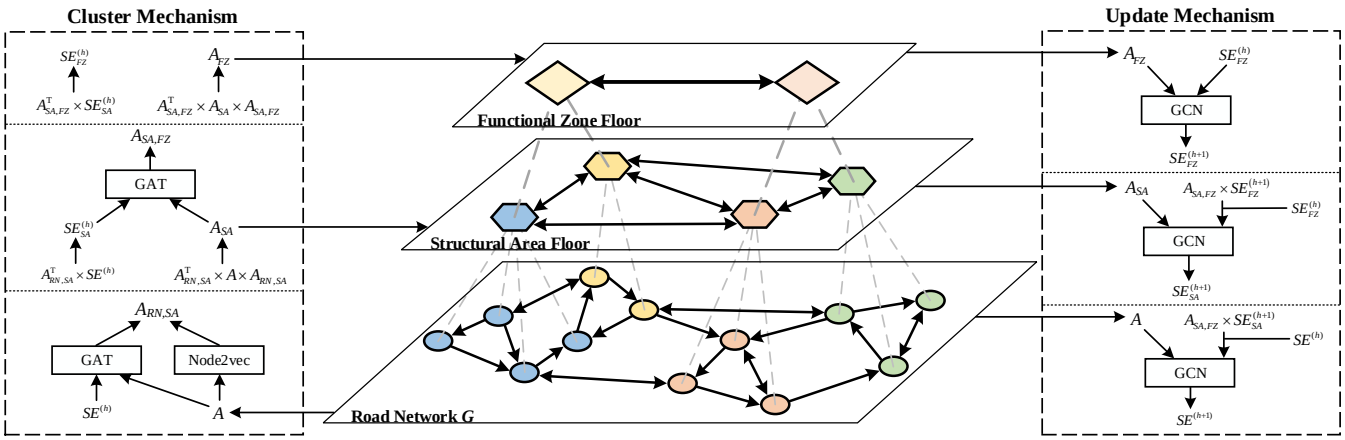


Fig. 6. Structure of hierarchical mapping network spatial module. Cluster mechanism is responsible for abstracting the original road network to extract the features of functional structure of and the long-distance dependence between nodes. The update mechanism is responsible for updating the node information of each floor from top to bottom, and feeding back the abstract features to the original nodes.

corresponding relationship matrix of the original road network to the structural area floor:

$$A_{RN,SA} = \text{softmax}\left(M_1 \odot IE_1\right), \tag{3}$$

where $\odot$ denotes the Hadamard product which is the element-wise multiplication between two matrices, and $\text{softmax}(\cdot)$ is the standard softmax function which normalizes each column of the matrix $A_{RN,SA}$. Next, utilize $A_{RN,SA}$ to calculate the features matrix of structural area floor:

$$SE_{SA,t_i} = A_{RN,SA}^{\mathrm{T}} \cdot SE_{t_i}, \tag{4}$$

where $SE_{SA,t_i} \in \mathbb{R}^{n_{SA} \times d_2}$ $(i \in \{1, 2, \ldots, \tau_x\})$ is the features matrix of the structural area floor, d_2 is also a constant. Moreover, the weighted adjacency matrix $A_{SA} \in \mathbb{R}^{n_{SA} \times n_{SA}}$ of structural area floor can also be calculated by similar method:

$$A_{SA} = A_{RN,SA}^{\mathrm{T}} \cdot A \cdot A_{RN,SA}, \tag{5}$$

2) Construct Functional Zones: This part elaborates the method of building a functional zone floor. This floor is also composed of virtual nodes, which is based on the structure area. Its main function is to calculate the information interaction between functional zones and capture the spatial features and the dependence between remote nodes.

A similar method is used in Eq. (4) to learn features of functional zone floor using the graph aggregation algorithm on structural area floor. Given the $A_{SA,FZ} \in \mathbb{R}^{n_{SA} \times n_{FZ}}$ mapping matrix from structural area floor to functional zone floor. $A_{SA,FZ}\left[v_{SA}, v_{FZ}\right]$ denotes the conditional probability that a structural area node v_{SA} belongs to a functional zone node v_{FZ}. Use the GAT network to calculate $A_{SA,FZ}$:

$$M_2 = \text{GAT}\left(SE_{SA,t_i}, A_{SA}\right), \tag{6}$$

$$A_{SA,FZ} = \text{softmax}(M_2), \tag{7}$$

where $M_2 \in \mathbb{R}^{n_{SA} \times n_{FZ}}$ represents the hard mapping matrix and the softmax function performed by columns of the matrix to derive the soft mapping matrix $A_{SA,FZ}$. In this way, the features matrix of functional zone floor can be set as the linear combination of the features matrix of structural area floor:

$$SE_{FZ,t_i} = A_{SA,FZ}^{\mathrm{T}} \cdot SE_{SA,t_i}. \tag{8}$$

With SE_{FZ,t_i}, the adjacency matrix of functional zone floor is calculated by the ReLU activation function to prevent the gradient from disappearing during training:

$$A_{FZ} = \text{ReLU}(SE_{FZ,t_i} \cdot SE_{FZ,t_i}^{\mathrm{T}} - \omega), \tag{9}$$

where A_{FZ} is the weighted adjacency matrix for graph of functional zone floor and ω is a scaling parameter.

3) Top-down Update Mechanism: In the above work, how to construct the features representation and adjacency matrix of each floor is described. In this part, the process of updating information and features from top to bottom is illustrated.

First, the update mechanism in functional zone floor is introduced. The features matrix needs to be updated and prepared for the information transmitted to the lower floor. The low complexity Graph Convolutional Network (GCN) based on Chebyshev polynomials [41] is used to update the feature embedding:

$$SE_{FZ,t_i}^{(h+1)} = \text{GCN}(SE_{FZ,t_i}^{(h)}, A_{FZ}), \tag{10}$$

where A_{FZ} is the computed adjacency matrix for functional zone nodes. On account of the A_{FZ} is not a binary matrix, GAT is not applicable to updating feature embedding in this case [37]. Then, send the functional zone embedding to the lower floor for updating structural area embedding:

$$\widetilde{SE}_{SA,t_i}^{(h)} = SE_{FZ,t_i}^{(h)} + g_{FZ,SA} \odot \left(A_{SA,FZ} \cdot SE_{FZ,t_i}^{(h+1)}\right), \tag{11}$$

$$g_{FZ,SA} = \sigma\left(\left(SE_{SA,t_i}^{(h)} || \left(A_{SA,FZ} \cdot SE_{FZ,t_i}^{(h+1)}\right)\right) \cdot w_1\right), \tag{12}$$

where $g_{FZ,SA}$ is a gate monitor which is similar to a weight matrix controling the information passing from functional zone floor to structural area floor, σ is the sigmoid function and w_1 is a trainable parameter vector.

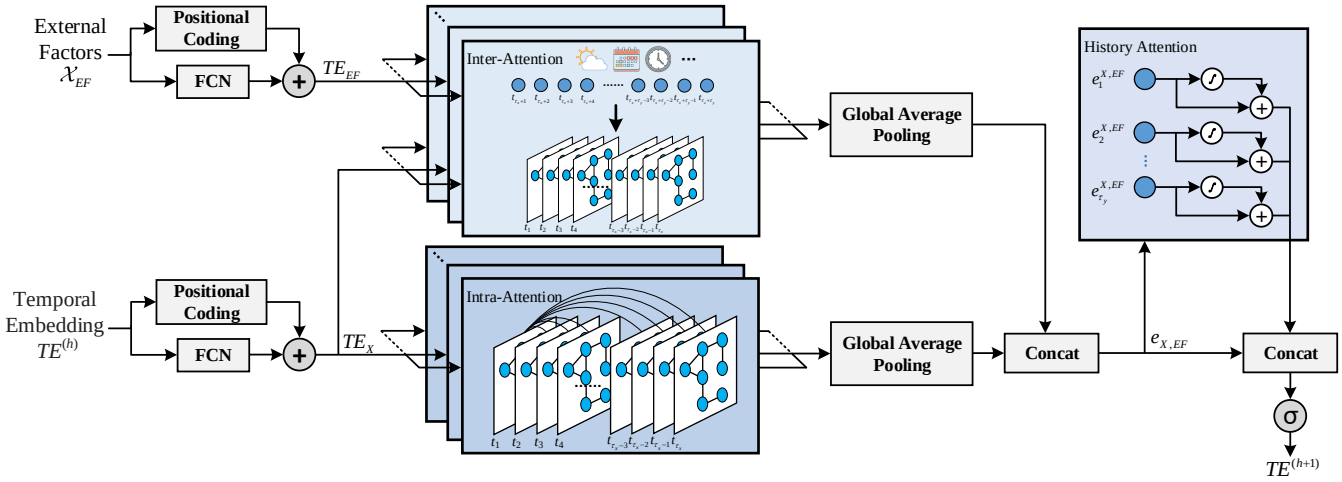


Fig. 7. Structure of Interactive Attention Temporal Module. Intra-attention is mainly used to calculate the interaction of internal elements of traffic data sequence. Inter-attention can calculate the impact of external factors on traffic data. Lastly, use the history attention to calculate the impact of historical data on future data.

At the structural-area-level, model first updates its own feature embedding with GCN:

$$SE_{SA,t_i}^{(h+1)} = \mathrm{GCN}\left(\widetilde{SE}_{SA,t_i}^{(h)}, A_{SA}\right), \tag{13}$$

where A_{SA} is the weighted adjacency matrix calculated by Eq. 5. Similarly, pass the updated information to the next floor:

$$\widetilde{SE}_{t_i}^{(h)} = SE_{t_i}^{(h)} + g_{SA,RN} \odot \left(A_{RN,SA} \cdot SE_{SA,t_i}^{(h+1)}\right), \tag{14}$$

$$g_{SA,RN} = \sigma\left(\left(SE_{t_i}^{(h)} || \left(A_{RN,SA} \cdot SE_{SA,t_i}^{(h+1)}\right)\right) \cdot w_2\right), \tag{15}$$

where $g_{SA,RN}$ is a gate monitor that controls the information passing from structural area floor to original road network, and w_2 is also a trainable parameter vector.

Finally, utilize a GAT to update the relation between segment nodes as follows

$$SE_{t_i}^{(h+1)} = \mathrm{GCN}\left(\widetilde{SE}_{t_i}^{(h)}, A\right), \tag{16}$$

where A is the adjacency matrix for the nodes of road network.

B. Interactive Attention Temporal Module

The module uses the interactive attention mechanism to fuse external factors and traffic features to calculate the impact of external factors on traffic, and employs the intra-attention mechanism and historical attention mechanism to extract the features of traffic data from the temporal dimension as shown in Fig. 7. The following content describes the content of the module and the calculation process.

1) Intra-attention Mechanism: Before calculating the interaction between different data sequences, the intra-attention mechanism should be applied alone to capture the internal correlation of features. In this paper, self-attention [42] is introduced to implement this mechanism instead of RNNs. The aim is to reduce the computational complexity and prevent the loss of location information of sequence. The following details the interaction of the algorithm processing data within the sequence.

In order to facilitate the calculation, a fully connected network (FCN) is employed to adjust the input data dimension. The input data includes traffic data and external factors which are recorded as embedding $TE_X^{temp} \in \mathbb{R}^{\tau_x \times d}$ and $TE_{EF}^{temp} \in \mathbb{R}^{\tau_y \times d}$ respectively after adjustment, where d is a constant.

In self-attention, in order to take the positional information into account, it is necessary to encode the data sequence by the position of each element in the sequence. The positional coding method mentioned in the transformer [42] is employed,

$$TE^{pos}(position, 2j) = \sin\left(\frac{position}{1000^{\frac{2j}{d}}}\right), \tag{17}$$

$$TE^{pos}(position, 2j+1) = \cos\left(\frac{position}{1000^{\frac{2j}{d}}}\right), \tag{18}$$

where $position$ is the position index of the element in the sequence, and $2j$ and $2j+1$ represent odd and even numbers in the sequence, respectively. Finally, the embedding of input data is obtained by adding the adjusted dimension data matrix and the location coding matrix as

$$TE_X = TE_X^{temp} + TE_X^{pos} \in \mathbb{R}^{\tau_x \times d}, \tag{19}$$

$$TE_{EF} = TE_{EF}^{temp} + TE_{EF}^{pos} \in \mathbb{R}^{\tau_y \times d}. \tag{20}$$

As mentioned above, data is encoded by self-attention mechanism after being transformed into embedding. Generally, the attention mechanism is to map a query and a set of key-value pairs to an output. The output result is the weighted sum of values, and the weight matrix is the inner product of query and key normalizing by the softmax function. In self-attention, the query Q, key K and value V are linear projections from the same input embedding matrix. The self-attention function is defined as

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^{\mathrm{T}}}{\sqrt{d_k}}\right)V, \tag{21}$$

where d_k is the second dimension of Q and K. The inner product of Q and K is scaled by $\sqrt{d_k}$ to avoid the small gradient caused by the large value of dot product.

To prevent overfitting, on the basis of self-attention, a multi-head attention mechanism is adopted to encode the input sequence. The input data embedding is projected into different n_{sub} subspaces through linear mapping, and the corresponding set of matrices $\{Q_m, K_m, V_m\}$ is generated in each subspace, where queries $Q_m = TE \cdot W_m^Q$, $W_m^Q \in \mathbb{R}^{d \times d_k}$, keys $K_m = TE \cdot W_m^K$, $W_m^K \in \mathbb{R}^{d \times d_k}$, and values $V_m = TE \cdot W_m^V$, $W_m^V \in \mathbb{R}^{d \times d_v}$, $m \in \{1, 2, \ldots, n_{sub}\}$. Finally, concatenate all attention heads together and project concatenated result to be the intra-encoding embedding as

$$\begin{aligned} TE_{intra} &= \text{MultiHead}(TE) \\ &= \text{Concat}(head_1, head_2, \ldots, head_{n_{sub}}) W_{intra} \end{aligned}, \tag{22}$$

where $head_m = \text{Attention}(Q_m, K_m, V_m)$, $W_{intra} \in \mathbb{R}^{dn_{sub} \times d_{intra}}$, and m ($m \in \{1, 2, \ldots, n_{sub}\}$) is the index of the heads. Set $d_k = d_v = d_{intra}/n_{sub}$.

It is not necessary to calculate the intra-attention for external factor sequence, because most of the external factors are objective elements, and traffic data does not adversely affect those factors. Given traffic embedding of traffic data TE_X, the intra-encoding embedding are calculated as $TE_X^{intra} = \text{MultiHead}(TE_X)$

2) Inter-attention Mechanism: In order to calculate the impact of external factors on traffic, that is to calculate the impact between different data sequences. The inter-attention mechanism is used to realize this function.

In the self-attention, the query, key and value are projected from the same data sequence to capture the internal interaction of the same sequence. In the inter-attention of this module, the query is projected from the external factors embedding, while key and value are projected from the traffic data embedding. Given the input sequences embedding TE_X and TE_{EF}, the inter-encoding embedding are calculated as follows,

$$\begin{aligned} TE_{X,EF}^{inter} &= \text{MultiHead}(TE_{EF}, TE_X) \\ &= \text{Concat}(head_1, head_2, \ldots, head_{n_{sub}}) W_{inter} \end{aligned}, \tag{23}$$

where $head_m = \text{Attention}(Q_m^{EF}, K_m^X, V_m^X)$, $Q_m^{EF} = TE_{EF} \cdot W_m^{Q_{EF}}$, $K_m^X = TE_X \cdot W_m^{K_X}$, and $V_m^X = TE_X \cdot W_m^{V_X}$.

To reduce the number of parameters and overfitting, the final encoding embedding of integration of traffic data and external factors are calculated by applying the global average pooling operator on the intra-encoding and inter-encoding results, then concatenate them as

$$e_{X,EF} = \text{Concat}\left(\text{pool}\left(TE_X^{intra}\right), \text{pool}\left(TE_{X,EF}^{inter}\right)\right), \tag{24}$$

where the pool is the global average pooling operator. After being processed by the historical attention mechanism described in detail below, vector $e_{X,EF}$ can be added to itself to obtain the final time embedding.

3) History Attention Mechanism: For the purpose of calculating the impact of historical traffic data on forecasting data, history attention is used to extract historical representation, i.e., calculating the weighted sum of historical data. Implementing historical attention mechanism on $e_{X,EF} = \{e_1^{X,EF}, e_2^{X,EF}, \ldots, e_{\tau_y}^{X,EF}\}$ generated from Eq. (24), the history attention representation $history$ is calculated as the weighted sum of embedding as

$$history = \sum_{i=1}^{\tau_x} \alpha_i e_i^{X,EF}. \tag{25}$$

The weights $\alpha = \{\alpha_1, \alpha_2, \ldots, \alpha_{\tau_x}\}$ is obtained by following:

$$\alpha_i = \text{softmax}(\lambda_i) = \frac{e^{{\lambda_i}^{\mathrm{T}} \cdot b_c}}{\sum_j e^{{\lambda_j}^{\mathrm{T}} \cdot b_c}}, \tag{26}$$

$$\lambda_i = \tanh(W_h \cdot e_i + b_h), i \in \{1, 2, \ldots, \tau_y\}, \tag{27}$$

where b_c represents a trainable history context vector.

The purpose of history attention is to extract the history representation information. Due to the spatial-temporal correlation of traffic data, traffic prediction is not only affected by the surrounding areas, but also affected by historical data.

The encoding vectors and historical attention vectors from traffic data and external factor sequence are concatenated together as

$$\tilde{e} = \sigma\left(\text{Concat}\left(e_{X,EF}, history\right)\right), \tag{28}$$

where σ is the sigmoid function.

Finally, the dimension of $\tilde{e}$ is adjusted through the full connection network (FCN) to obtain the final temporal embedding TE.

C. Gated Fusion Module

The module applies the gated mechanism to fuse the spatial embedding SE and temporal embedding TE from the above two modules to obtain the spatial-temporal feature embedding FE. As shown in Fig. 5(b), the fusion process is as follows:

$$FE^{(h)} = SE^{(h)} \odot g + TE^{(h)} \odot (1 - g), \tag{29}$$

$$g = \sigma\left(SE^{(h)} \cdot W_g^{SE} + TE^{(h)} \cdot W_g^{TE} + b_g\right), \tag{30}$$

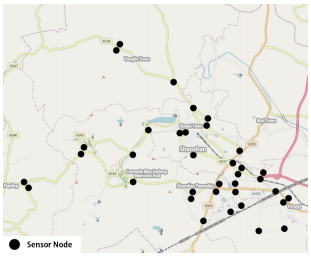
where $\odot$ denotes the element-wise product, g is the gate which controls the respective fusion degree of spatial and temporal features, and W_g^{SE}, W_g^{TE}, b_g are trainable parameters.

D. Loss Function

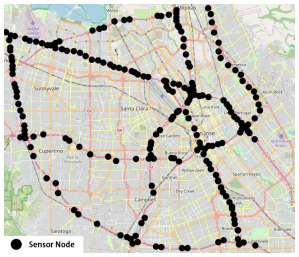
Loss function is used to calculate the difference between the predicted value and the true value in each training, and update the trainable parameters according to the difference by back propagation. In HMIAN model, the mean square error (MSE) is used as the loss function. The loss function is defined as

$$\text{MSE} = \frac{1}{\tau_y} \sum_{i=1}^{\tau_y} (Y_{t_i} - X_{t_i})^2, \tag{31}$$

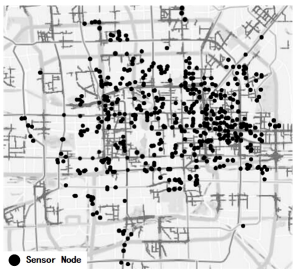
where Y_{t_i} and X_{t_i} are the prediction value and real traffic data respectively in t_i time step, and τ_y is the time steps of prediction results.



(a) Node distribution in Shaoshan.



(b) Node distribution in PeMS.



(c) Node distribution in Beijing.

Fig. 8. Distribution of sensor nodes in three data sets. (a) is the distribution of nodes in Shaoshan dataset, with a total of 36 nodes. (b) is the distribution of nodes in the PeMS dataset, with a total of 325 nodes. (c) is the the distribution of nodes in the Beijing dataset, with a total of 502 nodes.

TABLE II
DESCRIPTION OF THE DATASET

Dataset	#Nodes	#Edges	#TimeSteps	AVG	STD
Shaoshan	36	62	16128	33.47	11.65
PeMS	325	438	52116	58.89	13.48
Beijing	502	742	17568	37.82	20.28

IV. EXPERIMENT

A. Datasets

1) Traffic Datasets: This paper verifies the performance of HMIAN on three real datasets: the transport network dataset provided by Shaoshan scenic spot, the public transport network dataset PeMS and the Beijing traffic speed dataset provided by Baidu Map [43].

- Shaoshan traffic dataset is the traffic flow information collected through cameras arranged at key intersections. There are 36 sensor nodes in total, and the time range is from December 20, 2020 to February 13, 2021.
- PeMS dataset is the average speed information from the Bay Area of California. These data are collected from the California transportation administration performance measurement system (PeMS). There are 325 sensor nodes in total, and the time range is from January 1, 2017 to June 30, 2017.
- Beijing traffic dataset is the average speed information from the 6th ring road of Beijing. 502 nodes in this data set were selected as the experimental data, and the time range is from April 1, 2017 to May 31, 2017.

Take 5 minutes as a time point to count the traffic data in this time period, that is, there are 288 time points in the traffic data of a day. All data are normalized by Z-score algorithm. The distribution of sensor nodes in these data sets is shown in Figure 8. For specific descriptions of the three data sets, refer to Table II.

2) External Factors Datasets: The dimension of external factors data is 8, including which day of the week the data is on, time period, holiday, average temperature (°F), minimum temperature (°F), maximum temperature (°F), average wind speed (knots), visibility (MI). Among them, the meteorological data were collected by meteorological observation stations distributed around Bay Area and Shaoshan, and the data are provided by Convergent Weather. The data length is consistent with Shaoshan, PeMS and Beijing data set individually.

B. Experiment Settings

In this paper, the above data set is divided into training set, verification set and test set according to the ratio of 7:1:2. According to the rules of short-term traffic flow prediction and previous research works [32], [33], The historical traffic data of 12 consecutive time steps in one hour is used to predict the data of 12 consecutive time steps in the next hour. The final experimental results on each data set are the average of ten experimental results.

The experiment is carried out in the following hardware configuration. An Intel (R) Core (TM) i7-10700F CPU @ 2.90 GHz and NVIDIA GeForce RTX 2070 SUPER GPU 8GB card.

The hyperparameter settings are as follows: In order to ensure the efficiency of model training, Adam optimizer is used to train the model, and the learning rate is 0.01. To fully train all models, the batch size of input data is 32 and the training epoch is 100. The hyperparameters involved in NMIAN mainly include a scaling parameter ω in hierarchical mapping structure, the number of attention heads n_{sub} and the dimensionality d of each attention head in the Interactive Attention mechanism. These parameters are adjusted on the validation set to achieve the best performance of the HMIAN model (ω=0.5, n_{sub}=8, d=64).

The experimental results are measured by Mean Absolute Error (MAE), Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE):

$$\mathrm{MAE} = \frac{1}{\tau_y}\sum_{i=1}^{\tau_y}|Y_{t_i} - X_{t_i}|, \tag{32}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{\tau_y}\sum_{i=1}^{\tau_y}(Y_{t_i} - X_{t_i})^2}, \tag{33}$$

$$\mathrm{MAPE} = \frac{100\%}{\tau_y}\sum_{i=1}^{\tau_y}\left|\frac{Y_{t_i} - X_{t_i}}{Y_{t_i}}\right|, \tag{34}$$

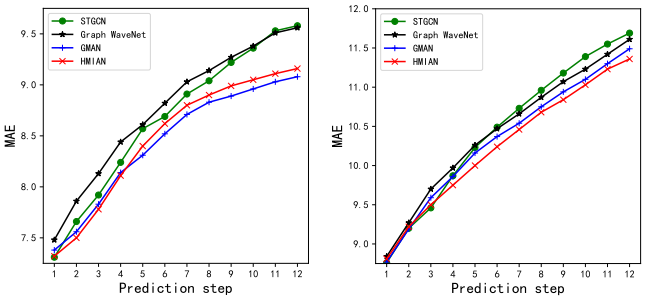
where Y_{t_i} and X_{t_i} are the prediction result and real traffic data separately in t_i time step, and τ_y is the number of time steps of prediction.

C. Results and Analysis Compared with Baseline Methods

This experiment is mainly to compare the HMIAN model with the baseline model to verify the performance of the model. The following baseline models are mainly from the

TABLE III
COMPARISON OF PREDICTION RESULTS BETWEEN HMIAN AND BASELINE MODEL ON SHAOSHAN, PEMS AND BEIJING DATA SETS.

Dataset	Metric	ARIMA	LSTM	DCRNN	STGCN	Graph WaveNet	GMAN	HMIAN
Shaoshan	MAE	10.33	10.16	9.98	9.58	9.56	**9.08**	9.16
	RMSE	16.85	16.39	16.09	15.37	15.45	14.98	**14.77**
	MAPE(%)	17.25	16.90	16.36	16.15	15.81	15.74	**15.62**
PeMS	MAE	14.77	13.40	13.02	11.69	11.61	11.49	**11.36**
	RMSE	23.88	23.11	22.41	20.04	20.73	19.62	**19.54**
	MAPE(%)	17.86	17.28	16.69	15.12	**14.93**	15.07	14.96
Beijing	MAE	13.05	12.61	11.94	12.11	11.38	11.70	**10.77**
	RMSE	16.83	16.43	15.73	16.12	15.20	15.56	**13.81**
	MAPE(%)	16.44	15.82	15.06	15.32	14.03	14.62	**12.41**



(a) Shaoshan. (b) PeMS.

(c) Beijing.

Fig. 9. Accuracy comparison of different prediction steps in three data sets.

research results of traffic flow prediction neighborhood in recent years:

- **ARIMA**: Autoregressive Integrated Moving Average model, which is a time series prediction and analysis method [44].
- **LSTM**: Long Short-Term Memory, which is a chain neural network for obtaining long-term dependence in sequence data [19].
- **DCRNN**: Diffusion Convolution Recurrent Neural Network, which utilizes diffusion convolution and sequence to sequence frameworks to capture spatial-temporal dependencies [33].
- **STGCN**: spatial-temporal Graph Convolutional Networks, which uses GCN to deal with the spatial features of traffic information [31].
- **GraphWaveNet**: Graph WaveNet framework, which constructs a self-adaptive adjacency matrix to preserve the hidden spatial dependencies [32].
- **GMAN**: Graph Multi-Attention Network, which is a self-coder composed of multiple spatial-temporal attention mechanisms [36].

The experimental results on the three data sets are shown in Table IV.

Compared with the traditional model-driven model, the deep learning method has higher prediction accuracy. This result proves that the deep learning method is more suitable for the prediction task of nonlinear traffic data. On the other hand, The prediction accuracy of LSTM used to process time series data is not as good as other spatio-temporal prediction models (DCRNN, STGCN, Graph WaveNet, GMAN, HMIAN), which indicats the spatial correlation in traffic data will affect the traffic prediction results, and explains that the main task of current research in traffic prediction field is extracting the spatio-temporal features of traffic data.

The results of the experiment show that the prediction accuracy of HMIAN model has reached the current research level. As the road network structure of the model becomes more complex and the number of nodes increases, the advantages of HMIAN model can be better reflected. As shown in Fig. 9, By comparing the prediction results of different time steps in the three data sets, HMIAN model has higher prediction accuracy in more complex road network, and can better ensure the prediction accuracy of 1 hour (12 time steps) in advance compared with other baseline models, which can provide more time for ITS to formulate strategies according to the traffic prediction.

D. Hierarchical mapping network performance validation

This part will conduct longitudinal comparative experiments to verify the performance of hierarchical mapping network in the real traffic data set. The comparison models are constructed by dismantling the hierarchical mapping network, and these models are used to replace the spatial feature extraction module of HMIAN

The comparison model is as follows:

- **Node2vec**: The algorithm is used to construct the hard mapping matrix between the first and second floors of the hierarchical mapping network, and extract the correlation between nodes based on the road network structure [40].

TABLE IV
PREDICTION BETWEEN HMIAN AND OTHER SPATIAL FEATURE EXTRACTION MODELS ON SHAOSHAN, PEMS AND BEIJING DATA SETS.

Dataset	**Metric**	Node2vec	GAT	GCN	1floor-HM	2floors-HM	**HMIAN**
Shaoshan	MAE	10.09	**8.84**	9.39	8.97	9.32	9.16
	RMSE	15.44	14.57	15.14	**14.32**	14.93	14.77
	MAPE(%)	17.25	16.90	16.36	**15.31**	15.81	15.62
PeMS	MAE	13.91	13.34	12.51	11.79	11.64	**11.36**
	RMSE	22.38	21.27	21.69	20.46	20.06	**19.54**
	MAPE(%)	17.58	16.76	16.90	15.75	15.21	**14.96**
Beijing	MAE	13.05	13.23	12.86	12.17	11.38	**10.77**
	RMSE	17.73	17.13	17.28	16.82	15.58	**13.81**
	MAPE(%)	16.85	16.49	16.17	15.59	14.71	**12.41**

TABLE V
THE ACCURACY COMPARISON OF DIFFERENT FUSION METHODS AND FUSING DIFFERENT EXTERNAL FACTORS.

Dataset	**External factors**	Data concatenating			**Inter-attention (HMIAN)**		
		MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)
Shaoshan	No external factors	9.46	15.34	15.91	9.16	14.77	15.62
	The day of week	9.41	15.27	15.87	9.02	14.64	15.53
	Time period	9.48	15.54	15.89	9.02	14.66	15.61
	Holiday	9.19	15.08	15.67	**8.76**	**14.36**	**15.31**
	Average temperature	9.33	15.42	15.91	9.10	14.76	15.42
	Minimum temperature	9.29	15.11	15.65	8.97	14.44	15.37
	Maximum temperature	9.23	15.28	16.73	9.15	14.64	15.49
	Average wind speed	9.37	15.41	15.88	9.04	14.69	15.48
	Visibility	9.30	15.49	16.07	9.22	14.83	15.59
	All factors	9.52	15.59	16.18	9.13	14.81	15.47
PeMS	No external factors	11.65	20.03	15.46	11.36	19.54	14.96
	The day of week	11.91	20.07	15.53	11.14	19.31	14.72
	Time period	11.87	19.98	15.59	11.25	19.34	14.76
	Holiday	11.67	20.02	15.54	11.13	**19.05**	**14.43**
	Average temperature	11.41	19.95	15.50	11.16	19.24	14.74
	Minimum temperature	11.35	19.87	15.38	**11.08**	19.17	14.66
	Maximum temperature	11.48	20.02	15.46	11.20	19.29	14.68
	Average wind speed	11.91	20.56	15.72	11.26	19.59	15.10
	Visibility	11.72	20.16	15.44	11.39	19.51	14.82
	All factors	11.81	20.04	15.50	11.30	19.68	15.91
Beijing	No external factors	11.02	14.27	13.31	10.83	13.95	12.79
	The day of week	10.94	14.16	13.18	10.67	13.73	12.57
	Time period	10.82	14.09	12.93	10.59	13.61	12.44
	Holiday	10.63	13.96	12.77	10.48	**13.38**	12.31
	Average temperature	10.57	14.26	12.91	**10.23**	13.47	**12.24**
	Minimum temperature	10.39	13.98	12.65	10.41	13.63	12.52
	Maximum temperature	10.46	14.28	12.92	10.62	13.72	12.57
	Average wind speed	10.87	14.37	12.88	10.54	13.75	12.66
	Visibility	11.22	14.58	13.62	11.04	14.21	13.18
	All factors	11.39	14.74	13.81	11.17	14.54	13.10

- **GAT**: Graph Attention Networks, which can be used to aggregate nodes in the road network and dynamically calculate the interaction between nodes [34].
- **GCN**: graph convolution network, which is the main algorithm to update the spatial features of nodes in each floor [29].
- **1floor-HM**: The model only uses GAT and GCN to update the node features of the original road network G.
- **2floors-HM**: this model The model only constructs the structural area floor (the second floor of) on the basis of the original road network G.

The results of this experiment show that the prediction accuracy of the basic model with strong universality such as GAT is higher, and the complex model will lead to poor prediction accuracy int the traffic data set with a simple road

network structure and few sensor nodes. In addition, more complex the road network is, more obvious the advantages of the hierarchical mapping network proposed in this paper are, namely, the prediction accuracy will improve with the increase of the number of layers of hierarchical mapping network. In PeMS data set and Beijing data set, HMIAN achieves the best accuracy compared with other models.

The above experimental results show that in the road network with more complex traffic structure, the interaction of structural areas and functional zones abstracted from the original road network will have a greater impact on the prediction results.

E. Comparative experiment of data fusion algorithms and external factors

In this experiment, the HMIAN model is compared longitudinally in terms of data fusion to verify the impact of different fusion methods on traffic prediction results. Based on the HMIAN, the model is adjusted as follows:

- **No fusion**: The temporal module does not fuse any external factors, but only extracts the time features of traffic data.
- **Data concatenating**: The model adopts the simplest fusion method. After the external factors are processed into embedding, they are directly spliced with the input data into a tensor.
- **Interactive attention**: The temporal module uses the interactive attention mechanism to calculate the effects of external factors.

In addition, in this experiment, different external factors are integrated to verify their influence on the prediction results.

The experimental results are shown in Table V. The results indicate that integrating external factors can indeed improve the prediction accuracy to a certain extent, but only taking external factors as a part of the input data tensor has little impact on the accuracy. The prediction accuracy is effectively improved by using interactive attention mechanism and integrating external factors.

From the type of external factors, in order to verify whether different external factors have a significant impact on the prediction results, an analysis of variance [45] is conducted according to the prediction results.

Firstly, the average value and variance of the prediction results considering different external factors are calculated, which are denoted as $\bar{y}_i$ and s_i $(i \in \{1, 2, \ldots, c_{EF}\})$, respectively. c_{EF} is the number of types of external factors. Then, the mean square of external factors (MSEF) and mean square error are calculated as follow:

$$\mathrm{MSEF} = \frac{\sum_{j=1}^{c_{EF}} n_j (\bar{y}_i - \bar{y})^2}{c_{EF} - 1}, \tag{35}$$

$$\mathrm{MSE} = \frac{\sum_{j=1}^{c_{EF}} \sum_{i=1}^{n_j} (y_{ij} - \bar{y}_i)^2}{(n_j - 1) \cdot c_{EF}}, \tag{36}$$

where n_j represents the number of experimental groups which adopt different fusion method separately. $\bar{y}$ is the average of all results, and y_{ij} is the prediction result of a certain group fusing a certain kind of external factor.

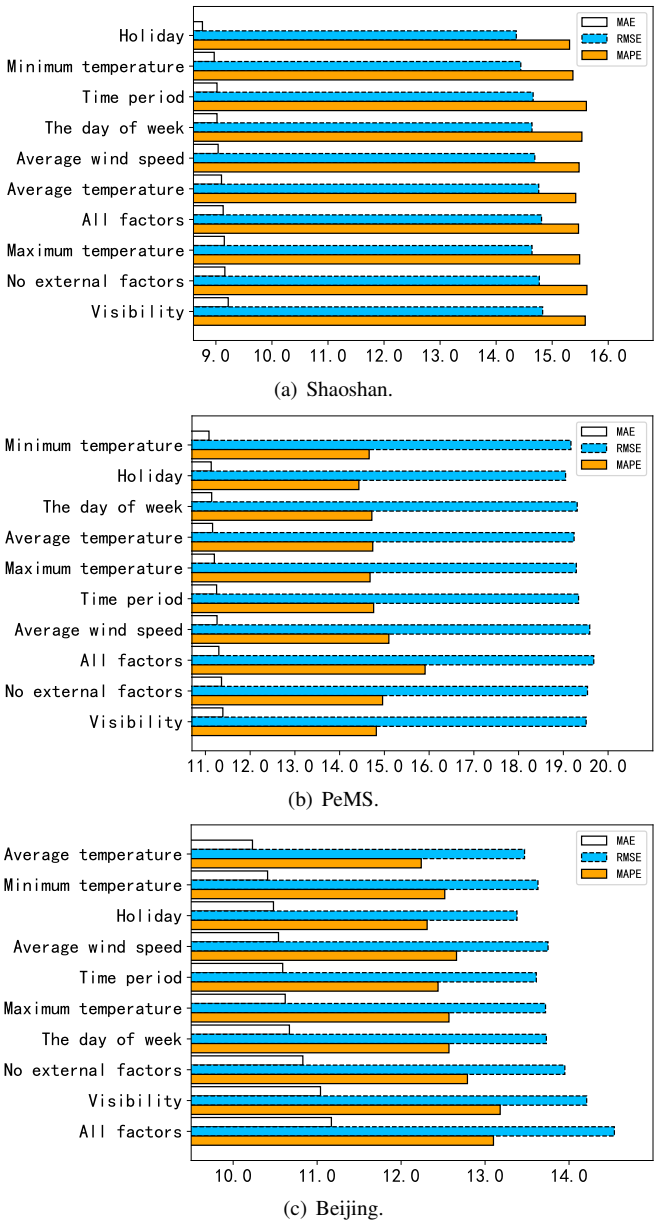


Fig. 10. The prediction accuracy of fusing external factors by interactive attention mechanism on the three data sets. The results are sorted by MAE.

F statistics is constructed to test the influence of different external factors on the prediction results. The calculation formula is as follows:

$$\mathrm{F} = \frac{\mathrm{MSEF}}{\mathrm{MSE}}. \tag{37}$$

Compare the calculated statistic F in the distribution critical value table [45]. In this experiment, the quantile α was set to 0.05. Since $\mathrm{F} > \mathrm{F}_\alpha\left(c_{EF} - 1, (n_j - 1) \cdot c_{EF}\right)$, it can be concluded that there are great differences between the effects of different external factors on the prediction results.

According to Fig. 10, the experimental results show that fusing the holiday and temperature factors can greatly improve the prediction accuracy. Since the collection site of Shaotshan

data set is located around the scenic spot, the holiday factor have a great impact on the prediction accuracy. The other two data sets have different degrees of influence from external factors due to their complex road network and functions. It can be seen that The influence of external factors on the prediction results of different types of regions will also be different. On the other hand, the experimental results show that fusing all factors can not achieve the best effect, and factors such as visibility may reduce the prediction accuracy.

V. Conclusion

In this paper, the HMIAN traffic prediction model is proposed, which can effectively aggregate the roads related to traffic information into functional zones, and extract the long-distance dependence relationship between roads in the traffic network by calculating the interaction between functional zones. When calculating the temporal correlation of data, the external factors affecting traffic are deeply integrated with data by using the attention mechanism such as interactive attention. Experiments were carried out on Shaoshan, PeMS and Beijing traffic data sets. The results show that HMIAN model has better prediction accuracy in more complex traffic networks with more sensor nodes. Furthermore, The longitudinal comparison experiment to verify the performance of hierarchical mapping network proves that the functional zones aggregated by roads have a great impact on the prediction in complex road network. And the comparative experiment of data fusion algorithms and external factors shows that compared with the general fusion method, the interactive attention mechanism can fuse traffic data and external factors more effectively, and improve the prediction accuracy. The result also verifies that factors, such temperature, holidays, have a greater impact on traffic prediction, and different regions are affected by external factors to different degrees due to their functional features. The following research will consider how to select appropriate external factors for fusion to improve the accuracy of traffic prediction.

References

[1] P. M. Santos, J. Rodrigues, S. B. Cruz, T. Lourenco, and J. Barros, "Portolivinglab: an iot-based sensing platform for smart cities," *IEEE Internet Things J.*, vol. 5, pp. 523–532, 2018.

[2] Z. Xiao, H. Fang, H. Jiang, J. Bai, V. Havyarimana, and H. Chen, "Understanding urban area attractiveness based on private car trajectory data using a deep learning approach," *IEEE Trans. Intell. Transp. Syst.*, pp. 1–10, 2021.

[3] J. Chen, K. Li, K. Li, P. S. Yu, and Z. Zeng, "Dynamic bicycle dispatching of dockless public bicycle-sharing systems using multi-objective reinforcement learning," *ACM Trans. Cyber-Phys. Syst.*, vol. 5, no. 4, 2021. [Online]. Available: https://doi.org/10.1145/3447623

[4] J. Wang and Q. Shi, "Short-term traffic speed forecasting hybrid model based on chaos–wavelet analysis-support vector machine theory," *Transp. Res. Pt. C-Emerg. Technol.*, vol. 27, pp. 219–232, 2013.

[5] 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. Pt. C-Emerg. Technol.*, vol. 54, pp. 187–197, 2015.

[6] S. Makridakis and M. Hibon, "Arma models and the box-jenkins methodology," *J. Forecast.*, vol. 16, no. 3, pp. 147–163, 1997.

[7] I. Okutani and Y. J. Stephanedes, "Dynamic prediction of traffic volume through kalman filtering theory," *Transp. Res. Pt. B-Methodol.*, vol. 18, no. 1, pp. 1–11, 1984.

[8] I. J. Chien and C. M. Kuchipudi, "Dynamic travel time prediction with real-time and historic data," *J. Transp. Eng.*, vol. 129, no. 6, pp. 608–616, 2003.

[9] J. Zheng and M. Huang, "Traffic flow forecast through time series analysis based on deep learning," *IEEE Access*, vol. 8, pp. 82 562–82 570, 2020.

[10] X. Peng, H. Zhu, J. Feng, C. Shen, and J. T. Zhou, "Deep clustering with sample-assignment invariance prior," *IEEE Trans. Neural Netw. Learn. Syst.*, no. 99, pp. 1–12, 2019.

[11] P. Hu, D. Peng, X. Wang, and Y. Xiang, "Multimodal adversarial network for cross-modal retrieval," *Knowledge-Based Syst.*, vol. 180, pp. 38–50, 2019.

[12] P. Hu, D. Peng, Y. Sang, and Y. Xiang, "Multi-view linear discriminant analysis network," *IEEE Trans. Image Process.*, vol. 28, no. 11, pp. 5352–5365, 2019.

[13] Y. S. Jeong, Y. J. Byon, M. M. Castro-Neto, and S. M. Easa, "Supervised weighting-online learning algorithm for short-term traffic flow prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 4, pp. 1700–1707, 2013.

[14] B. Jxa, X. Zhu, W. A. Dong, B. C. Jing, D. Vha, and A. Fz, "Short-term traffic volume prediction by ensemble learning in concept drifting environments - sciencedirect," *Knowledge-Based Syst.*, vol. 164, pp. 213–225, 2019.

[15] G. Filmon, Habtemichael, Mecit, and Cetin, "Short-term traffic flow rate forecasting based on identifying similar traffic patterns," *Transp. Res. Pt. C-Emerg. Technol.*, vol. 66, pp. 61–78, 2016.

[16] R. García-Ródenas, M. López-García, and M. Sánchez-Rico, "An approach to dynamical classification of daily traffic patterns," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 32, no. 3, pp. 191–212, 2016.

[17] Y. Lv, Y. Duan, W. Kang, Z. Li, and F. Y. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 865–873, 2015.

[18] Y. Jia, J. Wu, and Y. Du, "Traffic speed prediction using deep learning method," in *Proc. IEEE 19th International Conference on Intelligent Transportation Systems (ITSC'16)*, 2016, pp. 1217–1222.

[19] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with lstm," *Neural Comput.*, vol. 12, no. 10, pp. 2451–2471, 2000.

[20] I. Sutskever, O. Vinyals, and Q. Le, "Sequence to sequence learning with neural networks," in *Proc. 28th Annual Conference on Neural Information Processing Systems (NIPS'14)*, vol. 27, Montreal, CANADA, 2014.

[21] G. Dai, C. Ma, and X. Xu, "Short-term traffic flow prediction method for urban road sections based on space-time analysis and GRU," *IEEE Access*, vol. 7, pp. 143 025–143 035, 2019.

[22] D. Luo, D. Zhao, Q. Ke, X. You, L. Liu, and H. Ma, "Spatio-temporal hashing multi-graph convolutional network for service-level passenger flow forecasting in bus transit systems," *IEEE Internet of Things J.*, pp. 1–1, 2021.

[23] H. Jiang, Y. Zhang, Z. Xiao, P. Zhao, and A. Iyengar, "An empirical study of travel behavior using private car trajectory data," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 1, pp. 53–64, 2021.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[25] Z. Lv, J. Xu, K. Zheng, H. Yin, and X. Zhou, "Lc-rnn: A deep learning model for traffic speed prediction," in *Proc. 27th International Joint Conference on Artificial Intelligence (IJCAI'18)*, 2018.

[26] C. Chen, K. Li, S. G. Teo, X. Zou, K. Li, and Z. Zeng, "Citywide traffic flow prediction based on multiple gated spatio-temporal convolutional neural networks," *ACM Trans. Knowl. Discov. Data*, vol. 14, no. 4, 2020. [Online]. Available: https://doi.org/10.1145/3385414

[27] H. Yao, W. Fei, J. Ke, X. Tang, and J. Ye, "Deep multi-view spatial-temporal network for taxi demand prediction," in *Proc. 32nd AAAI Conference on Artificial Intelligence/ 30th Innovative Applications of Artificial Intelligence Conference/ 8th AAAI Symposium on Educational Advances in Artificial Intelligence*, 2018, pp. 2588–2595.

[28] Q. Liu, B. Wang, and Y. Zhu, "Short-term traffic speed forecasting based on attention convolutional neural network for arterials," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 33, no. 11, pp. 999–1016, 2018.

[29] J. Chen, K. Li, K. Li, P. S. Yu, and Z. Zeng, "Dynamic planning of bicycle stations in dockless public bicycle-sharing system using gated graph neural network," *ACM Trans. Intell. Syst. Technol.*, vol. 12, no. 2, 2021. [Online]. Available: https://doi.org/10.1145/3446342

[30] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc.*

30th Annual Conference on Neural Information Processing Systems (NIPS'16), 2016, pp. 3844–3852.
[31] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *Proc. 27th International Joint Conference on Artificial Intelligence (IJCAI'18)*, 2017.
[32] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," in *Proc. 28th International Joint Conference on Artificial Intelligence (IJCAI'19)*, 2019.
[33] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *Proc. 6th International Conference on Learning Representations (ICLR'18)*, 2018.
[34] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," 2017. [Online]. Available: https://arxiv.org/abs/1710.10903
[35] C. Tang, J. Sun, Y. Sun, M. Peng, and N. Gan, "A general traffic flow prediction approach based on spatial-temporal graph attention," *IEEE Access*, vol. 8, pp. 153 731–153 741, 2020.
[36] C. Zheng, X. Fan, C. Wang, and J. Qi, "Gman: A graph multi-attention network for traffic prediction," in *Proc. 34th AAAI Conference on Artifieial Inelligence*, 2020.
[37] N. Wu, X. W. Zhao, J. Wang, and D. Pan, "Learning effective road network representation with hierarchical graph neural networks," in *Proc. 26th ACM SIGKDD Conference on Knowledge Discovery & Data Mining (KDD'20)*, 2020.
[38] M. Li and Z. Zhu, "Spatial-temporal fusion graph neural networks for traffic flow forecasting," in *Proc. 35th AAAI Conference on Artificial Intelligence / 33rd Conference on Innovative Applications of Artificial Intelligence / 11th Symposium on Educational Advances in Artificial Intelligence*, 2021, pp. 4189–4196.
[39] Y. Shi, H. Feng, X. Geng, X. Tang, and Y. Wang, "A survey of hybrid deep learning methods for traffic flow prediction," in *Proc. 3rd International Conference on Advances in Image Processing (ICAIP'19)*, 2019.
[40] A. Grover and J. Leskovec, "Node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD'16)*, 2016.
[41] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *CoRR*, vol. abs/1609.02907, 2016. [Online]. Available: http://arxiv.org/abs/1609.02907
[42] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. 31st Annual Conference on Neural Information Processing Systems (NIPS'17)*, vol. 30, 2017.
[43] B. Liao, J. Zhang, C. Wu, D. McIlwraith, T. Chen, S. Yang, Y. Guo, and F. Wu, "Deep sequence learning with auxiliary information for traffic prediction," in *Proc. 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'18)*, 2018.
[44] G. Comert and A. Bezuglov, "An online change-point-based model for traffic parameter prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 3, pp. 1360–1369, 2013.
[45] R. A. Johnson and D. W. Wichern, *Applied Multivariate Statistical Analysis*. Tsinghua University Press, 2008.

Jingru Sun (Member, IEEE) received the Ph.D. degree in computer science and technology from Hunan University, China, in 2014.

She is currently an Associate Professor with the College of Computer Science and Electronic Engineering, Hunan University. She has published more than twenty articles. Her research interests include intelligent transportation, memristors, and its application to storage and neural networks.

Mu Peng is currently pursuing the master's degree with the College of Computer Science and Electronic Engineering, Hunan University.

His research interests include data mining and intelligent transportation technology.

Hongbo Jiang (Senior Member, IEEE) received the Ph.D. degree from Case Western Reserve University, Cleveland, OH, USA, in 2008. He is currently a Full Professor with the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China. He was a Professor with the Huazhong University of Science and Technology, Wuhan, China. His research focuses on computer networking, especially algorithms and protocols for wireless and mobile networks. He is the Editor of the IEEE/ACM TRANSACTIONS ON NETWORKING, an Associate Editor for the IEEE TRANSACTIONS ON MOBILE COMPUTING, and an Associate Technical Editor of the IEEE Communications Magazine.

Qinghui Hong received the B.S. and M.S. degrees in electronic science and technology from Xiangtan University, Xiangtan, China, in 2012 and 2015, respectively, and the Ph.D degree in computer system architecture from Huazhong University of Science and Technology, Wuhan, China, in 2019. He is currently an Assistant Professor with the College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China. His current research interests include memristive neural network and its application to Artificial Intelligence.

Yichuang Sun (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees from Dalian Maritime University, Dalian, China, in 1982 and 1985, respectively, and the Ph.D. degree from the University of York, York, U.K., in 1996, all in communications and electronics engineering.

He is currently a Professor of communications and electronics, the Head of the Communications and Intelligent Systems Research Group, and Head of the Electronic, Communication and Electrical Engineering Division, School of Engineering and Computer Science, University of Hertfordshire, U.K. He has published over 350 articles and contributed ten chapters in edited books. He has also published four text and research books: Continuous-Time Active Filter Design (CRC Press, USA, 1999), Design of High Frequency Integrated Analogue Filters (IEE Press, U.K., 2002), Wireless Communication Circuits and Systems (IET Press, 2004), and Test and Diagnosis of Analogue, Mixed-signal and RF Integrated Circuits: The Systems on Chip Approach (IET Press, 2008). His research interests include wireless and mobile communications, RF and analogue circuits, microelectronic devices and systems, machine learning, and deep learning.

Dr. Sun was a Series Editor of IEE Circuits, Devices and Systems Book Series from 2003 to 2008. He had been Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS from 2010 to 2011, from 2016 to 2017, and from 2018 to 2019. He is also Editor of ETRI Journal, the Journal of Semiconductors, and the Journal of Sensor and Actuator Networks. He was a Guest Editor of eight IEEE and IEE/IET journal special issues: High-frequency Integrated Analogue Filters in IEE Proceedings on Circuits, Devices and Systems in 2000, RF Circuits and Systems for Wireless Communications in IEE Proceeding on Circuits, Devices and Systems in 2002, Analogue and Mixed-Signal Test for Systems on Chip in IEE Proceeding on Circuits, Devices and Systems in 2004, MIMO Wireless and Mobile Communications in IEE Proceeding on Communications in 2006, Advanced Signal Processing for Wireless and Mobile Communications in IET Signal Processing in 2009, Cooperative Wireless and Mobile Communications in IET Communications in 2013, Software-Defined Radio Transceivers and Circuits for 5G Wireless Communications in the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II: EXPRESS BRIEFS in 2016, and the 2016 IEEE International Symposium on Circuits and Systems in the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS in 2016. He has also been widely involved in various IEEE technical committee and international conference activities.