Abstract—Traffic speed prediction is the key to many valuable applications, and it is also a challenging task because of its various influencing factors. Recent work attempts to obtain more information through various hybrid models, thereby improving the prediction accuracy. However, the spatial information acquisition schemes of these methods have two-level differentiation problems. Either the modeling is simple but contains little spatial information, or the modeling is complete but lacks flexibility. In order to introduce more spatial information on the basis of ensuring flexibility, this paper proposes IRNet (Transferable Intersection Reconstruction Network). First, this paper reconstructs the intersection into a virtual intersection with the same structure, which simplifies the topology of the road network. Then, the spatial information is subdivided into intersection information and sequence information of traffic flow direction, and spatiotemporal features are obtained through various models. Third, a self-attention mechanism is used to fuse spatiotemporal features for prediction. In the comparison experiment with the baseline, not only the prediction effect, but also the transfer performance has obvious advantages.

Index Terms—Traffic speed prediction, graph theory, deep learning, intersection reconstruction, transfer learning.

I. INTRODUCTION

With the development of the automobile industry and the improvement of people’s living standards, the number of automobiles has continued to increase. At the same time, the construction speed of infrastructure lags behind the growth rate of vehicles, which aggravates the problems of traffic congestion and low transportation efficiency. Road traffic speed is the average speed of vehicles on the road, and it is an important indicator for evaluating traffic conditions. Improving the utilization efficiency of infrastructure by predicting road traffic speed is a hot spot to solve traffic problems at present. For example, drivers can avoid slow road and choose better driving routes based on predicted road traffic speeds. On-board computers can adjust energy management strategies based on predicted road traffic speeds to save energy and reduce emissions. Traffic managers can manage road networks and allocate resources systematically based on predicted road traffic speeds [1]–[4].

Since the road traffic speed is closely related to the driving state of the vehicles on the road, road traffic speed is mainly affected by two aspects. (1) The impact of historical road traffic speeds. Although the road traffic speed changes all the time, it is impossible to completely separate from the historical road speed. (2) The impact of the traffic speed on adjacent roads. In a road network, different roads are connected by intersections. When a road is congested, it will directly affect the entry and exit of vehicles on adjacent roads. Therefore, the prediction of road traffic speed is a challenging task.

In the current research, traffic speed prediction methods are mainly divided into two categories: model-driven methods [5]–[8] that rely on theoretical mathematical models and data-driven methods that use statistical methods [9]–[11] and artificial intelligence algorithms [12]–[14]. Model-driven approaches use prior knowledge and physical properties to model traffic flow and predict road traffic speeds through traffic simulation. Therefore, the model-driven approach is highly interpretable. At the same time, less historical data is required when building the model [15]. However, model-driven methods require a large number of assumptions, such as time-invariant travel costs and homogeneous traveler route choices [16]. The accuracy of the model-driven approach depends on how accurately the model is built. However, in the process of building a high-precision model, different assumptions need to be adopted for different traffic conditions, which leads to an overly complex model.

Data-driven methods are data-dependent techniques that can break through the shackles of model-driven methods, and their effectiveness largely depends on the quality and size of traffic datasets. As data acquisition and storage become less difficult, data-driven approaches such as statistical methods and artificial intelligence algorithms are becoming more popular. In the early research works, data-driven methods for traffic speed prediction generally took the historical speed sequence of the target road as input, and found the change rule of traffic speed to predict the future traffic speed. Including statistical based methods such as ARIMA (Auto Regressive Integrated Moving Average Model) [11], KF (Kalman Filter Theory) [17], artificial intelligence algorithms such as machine learning algorithm SVR (Support Vector Regression) [12], and deep learning algorithm LSTM (Long and Short-term Memory) [18], [19] and so on. However, the above data-driven methods only focus on the historical data of the target road while the historical data of the adjacent roads...
are not utilized. To solve this problem, some works flattened the topology of the road network and add temporal information, so that the spatiotemporal information of traffic is converted into 2D images. In this way, the temporal and spatial correlations could be simultaneously obtained through CNN (convolutional neural network) [20]. However, due to the characteristics of CNN, compared with RNN (Recurrent Neural Network), which specializes in processing sequence data, CNN is relatively limited in its ability to mine time series information in data. This has prompted people to explore the fusion of multiple models. Some works used CNN to obtain spatial correlation and RNN to obtain temporal correlation, and obtain better prediction results [21]. Nevertheless, the transformed 2D image is difficult to reflect the topology of the real road network. Especially when the number of adjacent roads increases and the road network structure is complex, this processing method will produce huge deviations. And GCN (Graph Convolutional Network) shown excellent performance in spatial modeling of road networks in several recent works [22]–[24]. The limitation of GCN is that it requires full-batch training, so it takes up a lot of GPU memory during calculation. At the same time, GCN can only predict the speed of the road that participated in the training. Once the graph structure changes, the entire model needs to be rebuilt and retrained.

In general, existing traffic speed prediction methods tend to use flexible data-driven methods. The temporal and spatial features are extracted through a deep learning hybrid model to obtain higher prediction accuracy. Spatial feature extraction methods present a state of two-level differentiation. One is that all adjacent roads are tiled and each road can only be adjacent to two roads, which is far from the actual road structure and only retains part of the spatial information. The other is to consider the relationship between all of the roads, which brings huge computational pressure. At the same time, these two spatial feature extraction methods have the same problem. After the road network structure changes, the entire spatial feature extraction module needs to be redesigned and retrained, so the model has poor generality. This paper proposes IRNet (Transferable Intersection Reconstruction Network). The model defines a new method for obtaining spatial information of roads and combines spatiotemporal features to predict road traffic speed. The work and contributions of this paper are as follows:

1) A new multi-model fusion deep learning architecture is proposed, which simultaneously possesses good predictive performance and transfer ability. It is mainly divided into three modules according to functions. The data generation module obtains spatiotemporal information. The feature extraction module obtains spatiotemporal features through multiple deep learning models. The feature fusion module fuses multiple spatiotemporal features to predict the road traffic speed.

2) A new spatial feature extraction method called intersection reconstruction is defined. It simplifies the complex road network topology into the same road network structure. Therefore, when the road network structure changes, the prediction model does not need to change the structure. The expansion ability and transfer ability of the model are improved.

3) CNN has the characteristics of weight sharing and local connection, which is very suitable for extracting the spatial features of virtual intersections. LSTM is suitable for extracting spatial sequence features of traffic flow direction. By combining CNN and LSTM, the utilization of spatial information is improved, and highly representative spatial features can be learned.

4) The self-attention mechanism is introduced into the feature fusion module, which reflects the influence of different spatiotemporal features on the prediction of target road traffic speed, and is highly interpretable.

The rest of this paper is organized as follows. Section II reviews related work. Section III introduces some preliminary knowledge. Section IV introduces the method of model building in detail. Section V reports the experimental results. Section VI concludes the paper.

II. RELATED WORK

In model-based traffic speed prediction, methods using traffic flow theory are the focus of research. Traffic flow theory is based on mathematical and physical methods such as probability theory, mathematical statistics and calculus [25]. According to different research subjects, it can be mainly divided into microscopic models and macroscopic models. The research subject of the microscopic models is the vehicles on the road network, and the queuing phenomenon of the vehicles is evolved through the running process of the vehicles on the road network. The representative methods are the car-following model [5], [6] and the cellular automaton model [7], [8]. According to the theories of fluid dynamics and aerodynamics, the macroscopic models take the traffic flow as the main research body to establish the traffic model and solve the traffic evolution law. The representative methods are the velocity-density model [26] and the gas dynamics model [27]. In addition, the microscopic model and the macroscopic model are linked through the traffic wave theory, and a traffic flow model between microscopic and macroscopic is established. This model has common features of both microscopic model and macroscopic model [28], and its purpose is to combine the advantages of both while overcoming their respective limitations.

The derivation process of the traffic flow model is more rigorous, and the physical meaning of the model is clear. But some simplifying assumptions are required [29]. It also leads to the result that the traffic speed prediction accuracy of the model-based method largely depends on the accuracy of the traffic flow model. In order to improve the accuracy of the traffic flow model, Zheng et al. [30] established a driver’s emergency response mechanism based on the traffic flow model. Gao et al. [31] focused on the impact of low-visibility weather on car-following driving behavior. Geng et al. [32] studied the car-following driving behavior of different vehicle types under different environmental conditions. The addition of mechanisms can improve the accuracy of the traffic
flow model, and with it comes the complexity of the building process. Furthermore, it is very complicated to add additional measurement data after the model is established [33].

With the development of computer computing power and the development of various data storage devices, the data-driven approach has become a hot spot. Data-driven methods used in the field of traffic forecasting are mainly divided into statistical methods and artificial intelligence algorithms, and artificial intelligence algorithms can be subdivided into basic machine learning algorithms and deep learning algorithms. The statistical methods are to obtain the relationship between the variables through mathematical statistics method after the data is obtained. For traffic forecasting, it is usually assumed that the population obeys a certain distribution, and this distribution can be determined by some parameters. The model constructed on this basis is also called a parametric model. Such as HA (Historical Average Model) [9], AR (Auto Regressive Model) [34] and MA (Moving Average Model) [10]. With the development of the study, a number of more complex prediction methods with higher accuracy have emerged, including Kalman Filtering Model [17], Exponential Smoothing Model [35], etc. Among them, ARIMA (AutoRegressive Integrated Moving Average) considering the periodicity of speed data has a good effect in predicting traffic speed [11]. ARIMA has good theoretical interpretability and clear calculation structure, and is suitable for the prediction of road traffic speed. However, statistical methods fit data in a fixed functional form, usually only for simple problems. And the parameter tuning process is too complicated, resulting in poor online performance.

Compared with statistical methods, artificial intelligence algorithms can fit a variety of different functional forms and are more flexible to use. At the same time, the artificial intelligence algorithms have few prior assumptions about the input variables. Therefore, these methods have good fitting effect, and are more suitable for dealing with missing values and noisy data. However, artificial intelligence algorithms require more training data for fitting the objective function, and there is a higher risk of overfitting. Basic machine learning algorithms include SVR (Support Vector Regression) [12], [13], RF (Random Forest algorithm) [14], KNN (k-neighbor algorithm) [36], Bayesian model [37] and so on. The disadvantage of basic machine learning algorithms is that they lack the ability to analyze large-scale data. Therefore, the basic machine learning algorithm needs to select features manually, and the selection of features directly affects the prediction effect. Correspondingly, deep learning methods using two or more hidden layers can capture the feature expression in the input data through a multi-layer structure. Compared with the method of constructing features by artificial rules, deep learning algorithms can better describe the intrinsic information of data. In recent years, a great deal of literature on deep learning algorithms and the selection of features directly affects the prediction effect.

In deep learning algorithms, RNN (Recurrent Neural Network) can use internal storage units to process input sequences, thus giving neural networks the ability to learn time series. Due to the dynamic nature of traffic systems, RNN is particularly suitable for capturing the temporal evolution of traffic flow. LSTM (Long Short-Term Memory Network) is variants of RNN and have solved the vanishing gradient problem of RNN [20]. Ma et al. [41] used LSTM to capture long-term temporal dependencies in short-term vehicle speed prediction, and could determine the optimal time window for time series. Zhao et al. [42] integrated the origin destination correlation (ODC) in the LSTM network through a fully connected layer and a vector generator. However, the fully connected layer directly extracts the spatiotemporal features of all road networks, and does not focus on the target road that needs to be predicted at the end, and the spatial structure is not expressive enough. As a classic algorithm of computer vision, CNN (Convolutional Neural Network) is very good at capturing spatial information. Ma et al. [20] transformed the spatiotemporal information of traffic flow into 2D images according to the horizontal axis as time and the vertical axis as each road, and then obtained the spatiotemporal correlation of traffic flow simultaneously through CNN. Since CNN and LSTM have their own advantages in capturing spatiotemporal features, Wu et al. [43] proposed to obtain the spatial and temporal features of traffic flow through CNN and LSTM respectively, and finally combined all features for traffic flow prediction. However, CNN and LSTM do not interfere with each other, and the feature fusion is insufficient. Liu et al. [44] further refined the acquisition process of spatiotemporal features. The spatial features are captured by CNN, and then the captured spatial features are input into LSTM to capture the temporal features. However, there is a problem with the above methods. In the converted 2D image, the time axis is the same as the real situation, but the spatial axis is forced to flatten the complex spatial information. It does not reflect the topology of the road network, especially when the road network is complex.

GCN (Graph Convolutional Network), a feature extractor similar to CNN, is a deep learning network based on graph theory. The object of GCN is the graph describing non-European relations. Zhao et al. [22] used GCN to capture spatial features, and then used GRU to obtain temporal features, and fused the features to output prediction results. Lv et al. [24] introduced semantic information, encoded the correlation between roads through multiple graphs. Multiple GCNs were fused to capture spatial features, and then GRU was used to capture temporal features to obtain prediction results. GCN models the entire road network to obtain complete spatial information. But the disadvantage is that GCN is full-batch training, so it is difficult to apply to a large-scale road network. In the same way, GCN has poor flexibility and cannot deal with unseen nodes at all. When predicting the traffic speed of a road not involved in training, the graph needs to be restructured and the model needs to be retrained.

III. PRELIMINARY

A. Graph Theory

In graph theory, the graph is composed of vertices and edges connecting two vertices. This kind of graph usually uses vertices to represent objects, and an edge connecting two vertices indicates the relationship between the corresponding two
objects. The graph structure can effectively model and solve road network problems that need to consider the relationship between roads.

The road network can be defined as a graph \( G = (V, E) \), where \( V = \{v_1, v_2, \ldots, v_n\} \) is the set of vertices and each vertex \( v_i \) represents a road \( r_i \), \( E \subseteq \{e_{ij} = v_i v_j \mid v_i, v_j \in V\} \) is the set of edges and each edge \( e_{ij} \) represents the correlation between \( v_i \) and \( v_j \). In an undirected graph, \( v_i v_j = v_j v_i \). For road vertex \( v_i \), if there is an edge \( e_{ij} \in E \), road vertex \( v_j \) is a neighbor of road vertex \( v_i \), denoted as \( v_j \in N_i \). In other words, traffic flow on road \( r_i \) can flow to road \( r_j \) through the intersection. Let \( |V| = n \), the graph can be represented by an adjacency matrix \( A \in \mathbb{R}^{n \times n} \). The value of \( A_{i,j} \) reflects the adjacency relationship between \( v_i \) and \( v_j \) as follows:

\[
A_{i,j} = \begin{cases} 
1 & e_{ij} \in E \\
0 & e_{ij} \notin E 
\end{cases}
\]

B. Graph of Road

In this paper, the road is regarded as the vertex of the graph, the intersection connecting the road is the edge of the graph, and the value of the edge indicates the connection between the vertices. The average speed of the road is the feature of the vertex, and the average speeds of dual lanes of the same road are not the same, so this paper treats the dual lanes of the same road as two independent roads. Fig. 1(a) gives an example of a simplified road network and the roads in the figure are all one-way streets in the direction of the arrow. According to the description of graph theory in 3.1, the directed graph shown in Fig. 1(b) can be constructed by considering the roads as vertices.

According to the traffic flow theory, the vehicles on the road can be regarded as fluids and flow in the direction specified by the road. \( r_i \rightarrow r_j \) means that vehicles on \( r_i \) can merge into \( r_j \) through the intersection. If \( r_i \rightarrow r_j, r_j \) is the downstream adjacent road of \( r_i \), denoted as \( r_j \in D R_i \). Conversely, \( r_i \) is the upstream adjacent road of \( r_j \), denoted as \( r_i \in UR_j \). In addition, this paper defines that if \( r_j \in D R_i \) and \( r_k \in D R_j, r_k \) is the 2-order downstream adjacent road of \( r_i \), denoted as \( r_k \in D R_i^2 \), and so on. Similarly, \( r_i \) is the 2-order upstream adjacent road of \( r_k \), denoted as \( r_i \in UR_k^2 \). For the example in Fig. 1(b), \( r_6 \in D R_1^1, r_2 \in UR_6^2 \).

C. Problem

The study in this paper is to predict the average traffic speed of road nodes. There are three relevant parameters introduced here:

1. The length of historical sequence \( h \), \( h \) is an important parameter to measure the amount of time information. The average speed of the road \( r_i \) at time \( t \) is recorded as \( speed_i^t \), then \( S^t_i = \{speed_i^{t-h+1}, \ldots, speed_i^t\} \) is historical data sequence of the road \( r_i \) at time \( t \) of length \( h \).
2. Space width \( w \), \( w \) is an important parameter to measure the amount of spatial information. Space width in this paper refers to the order of adjacent roads centered on the target road, rather than the distance of the physical space.
3. Prediction horizon \( p \), \( p \) is a parameter to measure the predictive performance of the model, representing the span of the moment we predict based on the existing data. The average traffic speed of the target road with a prediction span of \( p \) is recorded as \( speed_{tar}^{t+p} \).

According to the above parameters, we define the problem as follows: For target road \( r_{tar} \), find all adjacent roads with the space width \( w \). At time \( t \), \( speed_{tar}^{t+p} \) is predicted according to the historical speed sequence of the adjacent roads and the target road with a length of \( t \).

IV. IRN METHOD

The model as shown in Fig. 2 is divided into three modules according to functions, the data generation module, the feature extraction module and the feature fusion module. The function of the data generation module is to generate input data from all the original road historical data according to the preset input parameters. The feature extraction module extracts the intersection features of each intersection through the convolution layer, and then extracts the temporal and spatial features of the adjacent roads through the Temporal-LSTM and Spatial-LSTM. The feature fusion module uses the self-attention mechanism to quantify the influence of spatio-temporal features of different orders on the final result and then outputs the prediction result.

A. Data Generation Module

The data generation module, as shown in Fig. 3(a), generates the input feature matrix of the target road network and the adjacent road networks. The key step in the generation of the input data of the adjacent road network is to simplify the complex road network topology into the intersection reconstruction operation of the same road network structure.

\[
S_{tar}^t = \{speed_{tar}^{t-h+1}, \ldots, speed_{tar}^t\}
\]

is the input data of the target road network, which is the historical speed sequence of the target road \( r_{tar} \) at time with a length of \( h \).

The input data of the adjacent road network is generated in two parts. Firstly, the complex road network topology is simplified through the intersection reconstruction operation, and the ordered sets of adjacent roads are obtained. Then, the sorted sets are transformed into the input data of the adjacent road network.
1) Intersection Reconstruction: There are many kinds of intersections in reality, as shown in Fig. 4. In most cases, the number of adjacent roads of different target roads is different. This paper proposes a method to reconstruct a variety of intersections into a class of intersection with the same structure without changing the characteristics of the original intersection as much as possible. The process of intersection reconstruction is divided into three steps. First, the roads at the intersection are numbered and sorted to ensure the similarity of intersection functions. Then normalize the number of roads connected to each intersection to ensure the similarity of the intersection structure. Finally, in order to reflect the road network structure after reconstructing the intersection, the sorting method of high-order adjacent roads is defined.

Since the functions and structures of upstream and downstream intersections are completely different, all upstream intersections should be reconstructed into one type of intersection and all downstream intersections should be reconstructed into another type of intersection. Here, the upstream intersection is used as an example to introduce how to reconstruct, and the downstream intersection is similar.

a) Sort The Adjacent Roads: This paper studies the prediction of traffic speed, focusing on the impact of adjacent roads on the future traffic speed of the target road. Therefore, a historical road speed sequence is taken as the feature vector of the road to reflect the speed characteristics of the road. The feature vector of road vertex \( v_i \) is \( F_i = [speed_1^i, ..., speed_\tau^i] \), where \( speed_t^i \) is the road speed of road vertex \( v_i \) at time \( t \), and \( \tau \) is the length of the historical speed sequence. After obtaining the feature vectors of all roads, the correlation of any two feature vectors can be calculated. When studying the speed of the road, the correlation of the road feature vector is approximately equal to the correlation of the road speed. This paper uses DTW (Dynamic Time Warping algorithm) [45] to calculate the correlation among feature vectors. Compared with the Euclidean distance calculated by corresponding points at the same position of the two feature vectors, DTW can map a point of one feature vector to multiple points of another feature vector. In other words, any point of the feature vector will find the point on the other feature vector with the smallest possible distance to calculate the distance. In order to evaluate the impact of the speed of the adjacent road on the speed of the target road, the similarity of the overall trend is more important than the similarity of each moment. DTW has the effect of stretching or compressing the sequence, so it is more suitable for measuring the correlation of the road feature vectors. The formula is as follows:

\[
DTW(A_1, B_j) = \sqrt{q |a_i - b_j|^q + \left( \min \left\{ \frac{DTW(A_{i-1}, B_{j-1})}{DTW(A_i, B_{j-1})}, \frac{DTW(A_{i-1}, B_j)}{DTW(A_i, B_{j-1})} \right\} \right)^q}
\]  

(2)

where \( A_i \) is the sequence \( (a_1, ..., a_i) \), \( |a_i - b_j|^q \) is the \( q \)-order distance between the \( i \)th point of \( A_i \) and the \( j \)th point of \( B_j \). Let \( q = 2 \) in this paper.

For any target road, sort the upstream adjacent roads according to Algorithm 1. It can be seen that the label of the adjacent road indicates the degree of correlation between the adjacent road and the target road. Adjacent roads with the same label number of different target roads play the similar structural roles in their respective intersections.
Algorithm 1 Sort the upstream adjacent roads of the target road (SORTUP)

Input: Target road $r_{tar}$; The set of 1-order upstream adjacent roads of the target road $UR_{tar}^{1}$.

Output: The ordered set of upstream adjacent roads of the target road $SUR_{tar}$.

1: Initializing $z \leftarrow 0$;
2: while $z < |UR_{tar}^{1}|$ do
3: \hspace{1em} $z \leftarrow z + 1$
4: \hspace{1em} Finding the $z$th road $r_{z}$ in the set of 1-order upstream adjacent roads, $r_{z} = UR_{tar}^{1}|z|$;
5: \hspace{1em} Calculating the correlation between $r_{z}$ and $r_{tar}$, and take the result as the label $l_{z}$ of $r_{z}$,
6: \hspace{1em} $l_{z} \leftarrow DTW (F_{t}, F_{tar})$.
7: Sorting the roads from small to large according to the label to get the sorted set of upstream adjacent roads of the target road $SUR_{tar}$.
8: return $SUR_{tar}$

b) Normalization of Intersection: By sorting the adjacent roads, various real intersections are abstracted into similar virtual intersections. The structures of these intersections are similar, but the number of connecting roads at each intersection is probably different. This paper stipulates that each intersection is fixedly connected $k$ adjacent roads to the target road. All virtual intersections are normalized according to Algorithm 2, and reconstructed intersections with completely consistent structure are obtained.

Algorithm 2 Normalization of upstream adjacent roads of the target road (NORMUP)

Input: Target road $r_{tar}$; The set of 1-order upstream adjacent roads of the target road $UR_{tar}^{1}$; The normative number $k$ of adjacent roads of the target road.

Output: The normalized set of upstream adjacent roads of the target road $NR_{tar}$. 

1: Sorting the upstream adjacent roads of the target road $SUR_{tar} = SORTUP (r_{tar}, UR_{tar}^{1})$;
2: if $|SUR_{tar}| > k$ then
3: \hspace{1em} $NR_{tar} \leftarrow$ the first $k$ roads in $SUR_{tar}$
4: else if $|SUR_{tar}| = k$ then
5: \hspace{1em} $NR_{tar} \leftarrow SUR_{tar}$
6: else
7: \hspace{1em} $NR_{tar} \leftarrow SUR_{tar}$ and the remaining $k - |SUR_{tar}|$ positions are supplemented with dumb points.
8: return $NR_{tar}$

c) Sort The High-Order Adjacent Roads: In order to obtain more spatial information near the target road, multiple virtual intersections are formed into a simplified road network. According to Algorithm 3, the $d$-order upstream adjacent roads of any target road can be transformed into an ordered set with $k^{d}$ elements.

Algorithm 3 Sort the $d$-order upstream adjacent roads ($d > 1$)

Input: Target road $r_{tar}$; The set of 1-order upstream adjacent roads of the target road $UR_{tar}^{1}$; The normative number $k$ of adjacent roads of the target road; The ordered set of $(d-1)$-order upstream adjacent roads of the target road $U_{tar}^{d-1}$, specially, $U_{tar} \leftarrow NORMUP (r_{tar}, UR_{tar}^{1}, k)$.

Output: The ordered set of $d$-order upstream adjacent roads of the target road $U_{tar}^{d}$.

1: Initializing $U_{tar}^{d} \leftarrow \emptyset$ and $z \leftarrow 0$
2: while $z < |U_{tar}^{d-1}|$ do
3: \hspace{1em} $z \leftarrow z + 1$
4: \hspace{1em} Finding the $z$th road $r_{z}$ in the ordered set of $(d-1)$-order upstream adjacent roads of the target road $U_{tar}^{d-1}$, $r_{z} = U_{tar}^{d-1}|z|
5: \hspace{1em} if $r_{z}$ is a dumb point do
6: \hspace{2em} $U_{tar}^{d} \leftarrow U_{tar}^{d-1} \cup DS$, where $DS$ is the set of $k$ dumb points
7: \hspace{1em} else
8: \hspace{2em} $U_{tar}^{d} \leftarrow U_{tar}^{d} \cup NORMUP (r_{z}, UR_{z}, k)$
9: return $U_{tar}^{d}$

The functions of the downstream intersection and the upstream intersection are different, but the structure is that one target road corresponds to multiple adjacent roads. So the same process can be used to get the ordered set of $d$-order downstream adjacent roads of the target road $D_{tar}^{d}$.

2) Convert Set to Input Matrix: As shown in Fig. 3(a), $U_{tar}, U_{1}^{1}, D_{1}^{1}, D_{w}$ are the ordered sets of adjacent roads of each order after the intersection reconstruction. All sets are transformed into matrices according to the rules shown in Fig. 3(b). For any $r_{i} \in InputSet$, $S_{i}^{t} = \{speed_{i}^{t-h+1}, \ldots, speed_{i}^{t}\}$ is the historical speed sequence of the target road $r_{i}$ at time $t$ with a length of $h$. Obtain the historical speed sequence of all roads in the set, and then stack them in the order of the ordered set to get a historical speed matrix $M$ corresponding to the set. And $M \in R^{m \times k}$, $m = k^{d}$, where $k$ is the normative number of the adjacent roads, $d$ is the order of the adjacent roads.

B. Feature Extraction Module

The historical data sequence of the target road only contains temporal information, while the historical data matrix of the adjacent roads contains spatiotemporal information. Therefore, the feature extraction module is divided into two sub-modules, one is the target road network, and the other is the adjacent road network.

1) Target Road Network: The structure of the target road network is shown in Fig. 5(a). The input data of the target road network is the historical speed sequence of the target road with a length of $h$, $S_{tar}^{t} = \{speed_{tar}^{t-h+1}, \ldots, speed_{tar}^{t}\}$, $S_{tar}^{t} \in R^{1 \times h}$. After the sequence passes through the LSTM module, the results of each LSTM unit are spliced together. Finally, a fixed length feature vector $trf^{d}$ of the target road is obtained through the fully connected layer, and the dimension of $trf^{d}$ is $d_{hid}$.

Recurrent Neural Network (RNN) as shown in Fig. 5(b) can use historical information to learn the current prediction
A task, which is more suitable for sequential data than other neural networks. In an ordinary RNN, there is only a simple structure, such as a tanh layer. So when the interval between the historical information and the current predicted position becomes larger, it is difficult for the RNN to learn the correlation between two steps far apart due to gradient vanishing or gradient exploding.

Whereas RNN have only one transfer state, LSTM as shown in Fig. 5(b) have two transfer states, a cell state $c_s$ and a hidden state $h_s$. Meanwhile, LSTM has three more gates (forget gate $g_f$, input gate $g_i$, output gate $g_o$) as shown in Fig. 5(c). The calculation formula for each gate is as follows:

$$g_f^t = \sigma(W_f \frac{x^t}{h^t-1} + b_f)$$

$$g_i^t = \sigma(W_i \frac{x^t}{h^t-1} + b_i)$$

$$g_o^t = \sigma(W_o \frac{x^t}{h^t-1} + b_o)$$

where $x^t$ is the data at time $t$ in the input sequence, $W_f, b_f$ are the weight parameters and bias parameters of the forget gate. $W_i, b_i$ are the weight parameters and bias parameters of the input gate. $W_o, b_o$ are the weight parameters and bias parameters of the output gate. $\sigma$ denotes the logistic sigmoid function. $c_s^t \in [-1, 1]$ is the updated value of the cell state. $c_s^t$ and $h_s^t$ are the weight parameters and bias parameters of the update neural network layers. $g_f^t, g_i^t$ and $g_o^t$ are respectively the value of the forget gate, the input gate and the output gate at time $t$. $g_f^t, g_i^t, g_o^t \in [0, 1]$. The forget gate $g_f^t$ is used to control the cell state of the previous state $c_s^{t-1}$. The input gate $g_i^t$ selects the input $x^t$. The output gate $g_o^t$ is used to control the cell state $c_s^t$ to output at the end. $c_s^t$ and $h_s^t$ are obtained as follows:

$$c_s^t = g_f^t \odot c_s^{t-1} + g_i^t \odot \tilde{c}_s^t$$

$$h_s^t = g_o^t \odot \tanh(c_s^t)$$

where $\odot$ is Hadamard product, which is the multiplication of the corresponding elements in the matrix. It can be seen from the formula that $c_s$ generally changes slowly, and $h_s$ is very different from the previous moment. Based on multiple gating mechanisms and two transit states, LSTMs are more capable of capturing long-term dependencies.

2) Adjacent Road Network: As shown in Fig. 6(a), the structure of the adjacent road network is divided into three parts: the convolutional layer to extract intersection features, the Temporal-LSTM to extract temporal features, and the Spatial-LSTM to extract spatial features. The input matrix of the $d$-order adjacent road is processed by the convolution layer and Temporal-LSTM to obtain the temporal feature $tf^d$ of the $d$-order adjacent road at time $t$. The sequence is formed by the temporal features of adjacent roads of each order according to the sequence of vehicle flow. The spatial feature of adjacent roads of each order is obtained through Spatial-LSTM. The spatial feature of the $d$-order adjacent road at time $t$ is recorded as $sf^d_{\star}$.

a) the convolutional layer: Known from the data generation module, the stacking order of the historical speed matrix from top to bottom is consistent with the order of the ordered set. Therefore, every $k$ row in the matrix is the historical speed data of an intersection, and the adjacent roads with the same relative position have similar roles in the intersection. Convolutional neural network extracts data features by sliding the convolution kernel on the data. The convolution kernel locally connects neurons and input data, and the weights of the convolution kernels are shared. These properties make convolutional neural networks particularly suitable for extracting the features of a single intersection at each moment in the input matrix. The convolution process can be expressed according to the structure of the intersection by the following formulas:

$$Conv_{\bar{U}, \star} = \sigma(W_{\bar{U}} * U M_{\star} + b_{\bar{U}})$$

where $\bar{U}$ is the updated value of the cell state. $\sigma$ denotes the logistic sigmoid function.
operation of the convolutional layer.

\[ Conv_{c,d}^{t} = \sigma(W_{c}^{t} \ast D M_{d}^{t} + b_{c}^{t}) \]  

(10)

where \( U M_{d}^{t} \) is the historical speed matrix of the \( d \)-order upstream adjacent road at time \( t \). \( W_{c}^{t}, b_{c}^{t} \) are the weight parameters and bias parameters of the \( c \)-th convolution channel of the upstream intersection. \( Conv_{c,d}^{t} \) is the output of the historical speed matrix of the \( d \)-order upstream adjacent road after passing through the \( c \)-th convolution channel. The convolution formula of the downstream adjacent road network is the same.

The specific operation of the convolutional layer is shown in Fig. 6(b). Taking the upstream road as an example, the processing method of the downstream road is the same. In order to make the convolution kernel exactly correspond to the data of one intersection at one moment, take \( W_{c}^{t} \in \mathbb{R}^{k \times 1} \). The stride of the convolution kernel is set to \([k, 1]\). It means the distance that the convolution kernel slides on the data after each convolution operation. It can be calculated \( Conv_{c,d}^{t} \in \mathbb{R}^{(m+k) \times h}, \ m = k^{d} \). At the same time, because the convolution kernel size and stride do not stretch the matrix in the time dimension, the length is still \( h \). The convolutional layer in this paper obtains different features of each intersection structure through multiple convolutional channels. Then the multi-channel intersection features are stacked together to construct the intersection feature matrix. Treat the column data of the feature matrix as the feature vector \( uif_{d}^{t} \) of the corresponding time, a sequence of intersection features \( UIF_{d}^{t} = [uif_{d}^{t-h+1}, \ldots, uif_{d}^{t}] \) can be obtained.

Since the features of the intersection structure are learned, the historical speed matrix of the upstream adjacent roads of different orders can extract the intersection features through the convolution layer of the same structure and the same parameters. It should be noted that the input matrices of different orders have different sizes, so \( uif \) of different orders have different dimensions.

### b) LSTM Layer

The core of both Temporal-LSTM and Spatial-LSTM is the basic LSTM structures as described in the target road network. The main difference between temporal LSTM and spatial LSTM is that the input data are different, so different features can be captured. At the same time, the two methods are slightly different in other parts except LSTM.

The input of the Temporal-LSTM is the sequence of intersection features obtained by the convolutional layer \( IF_{d}^{t} = [i f_{d}^{t-h+1}, \ldots, i f_{d}^{t}] \). The sequence length is \( h \), and each moment has \( c \times k^{d-1} \) elements. The structure of Temporal-LSTM is the same as the structure of the target road network. The outputs of all LSTM units are spliced together, and the fully connected layer outputs a temporal feature vector \( tf_{d}^{t} \) of the \( d \)-order adjacent roads at time \( t \) with a fixed length.

The speed of the road is closely related to the flow of vehicles in the road network, and the flow direction of the vehicles is unique. Therefore, the temporal feature vector \( tf_{d}^{t} \) of the adjacent roads can be formed into a spatial sequence in the order from high order to low order, \( TF^{d} = [tf_{d}^{1}, \ldots, tf_{d}^{h}] \). The spatial sequence is processed by LSTM, and the output of each LSTM unit is taken as the spatial feature vector \( sf_{d}^{\tau} \) of adjacent roads of the corresponding order. The dimensions of all spatial feature vectors are the same as the target road output vector \( trf^{\tau} \), which is \( d_{hid} \).

### c) Feature Fusion Module

In order to reflect the influence of the adjacent roads of each order on the target road, this paper combines the self-attention mechanism to quantify the relationship among the roads of each order. Through three different fully connected layers, the spatial feature vector \( sf_{d}^{\tau} \) can be mapped into three vectors respectively called Query \( Q_{d} \), Key \( K_{d} \), and Value \( V_{d} \). The value of the dot product of \( Q_{d} \) and \( K_{j} \) indicates the correlation between \( sf_{d}^{\tau} \) and \( sf_{j}^{\tau} \). Next, divide the value by the square root of the vector dimension \( d_{hid} \) and use the softmax function to get the attention weight. Finally, the attention weight is multiplied by \( V_{d} \). For computational efficiency, the vectors can be stacked as a matrix for calculation as shown in Fig. 7, \( Q \in \mathbb{R}^{(2d+1) \times d_{hid}}, K \in \mathbb{R}^{(2d+1) \times d_{K}}, \)
Fig. 8. (a) Scenarios for Sensor Deployment. (b) The data area selected in this paper.

\[ V \in \mathbb{R}^{(2d+1) \times d_V} \]. The formula is as follows:

\[
\text{Atten}(Q, K, V) = \text{softmax} \left( \frac{QK^T}{\sqrt{d_{hid}}} \right) V
\]  

(11)

where \( d_{hid} \) is the vector dimension for calculating attention weights, in this paper, \( d_Q = d_K = d_V = d_{hid} \).

Finally, the obtained result is flattened and the final output is obtained through the fully connected layer.

V. EXPERIMENT

A. Preparation

1) Source Dataset: This paper evaluates the model with the Caltrans Performance Measurement System (PeMS), a real-world traffic dataset. A total of 39,000 sensors are deployed on highway systems in all major urban areas of California, as shown in Fig. 8(a). A region of California was used, as shown in Fig. 8(b). And the data from January 1, 2020 to January 31, 2020 recorded every hour are used as the source data.

2) Data Preprocessing: The opposite lanes are regarded as two different roads, and there are 234 roads in the selected area. Match each road with the sensors installed on the road, and take the average speed of all sensors corresponding to a road to represent the speed of the road. The interval time is taken as 1 hour, that is to say, each road contains 744 historical speed records.

3) Evaluation: The effect evaluation method in this paper is Root Mean Square Percentage Error (RMSPE) and Mean Absolute Percentage Error (MAPE), and the formula is as follows:

\[
\text{RMSPE} = \sqrt{\frac{1}{m} \sum_{i=1}^{m} \left( \frac{y_i - \hat{y}_i}{y_i} \right)^2} \times 100%
\]  

(12)

\[
\text{MAPE} = \frac{1}{m} \sum_{i=1}^{m} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100%
\]  

(13)

where \( y_i \) is the actual value and \( \hat{y}_i \) is the predicted value.

B. Parameter

1) Model Parameter: In the data generation module, the parameter that need to be defined is the normative number \( k \). In this paper, \( k = 3 \), which means that an intersection contains one target road and three adjacent roads.

The parameters of the training process are shown in Table 1. The main components of the target road network are one LSTM layer and one fully connected layer, and the parameters are shown in Table 2. The main components of the adjacent road network are a convolutional layer, one LSTM layer and one fully connected layer of Temporal-LSTM, and one LSTM layer of Spatial-LSTM. The parameters are shown in Table 3.

2) Input Parameter: As mentioned in Section III, there are two important input parameters for the input data of the model. One is the space width \( w \), and the other is The length of historical sequence \( h \). According to the definition, it can be seen that the larger \( h \) and \( w \) are, the more spatiotemporal information is input into the model. In general, more spatiotemporal features can be captured from more spatiotemporal information, and predictions are more accurate. However, the calculation cost of the model increases accordingly, as shown in Table 4. The goal is to use as small \( w \) and \( h \) as possible to obtain a better prediction effect. This paper sets \( h \in [2, 3, 4, 5, 6] \) and \( w \in [1, 2, 3] \). The result for \( p \in [1, 2, 3, 4, 5] \) is shown in Fig. 9. When \( w \) remains constant and \( h \) increases, the result is shown in Figure 9(a-c). With the increase of \( h \), the overall speed prediction effect for the future time is
constantly getting better. Especially for the input with larger w, increasing h has a more obvious effect on the prediction result. In the same way, when h continues to increase, the prediction effect is not significantly improved and the amount of calculation is increased, so the experiments after this paper set h = 6. Conversely, when h is small, increasing w will lead to worsening of the effect. This is because in a short time, the traffic flow of the adjacent road far from the target road has less influence on the target road. In this paper, all 15 prediction results with h > 3 are stacked as shown in Fig. 9(d). As h increases, increasing w improves the prediction effect significantly. Due to hardware limitations, w of all subsequent experiments in this paper is set to 3.

C. Comparison with Baselines

For sake of evaluating the prediction performance of irnet, five benchmark methods are used. Among them, ARIMA can only input the historical speed sequence of the target road. The input of CNN, LSTM, SAES, ResMLP can be the input matrix of the target road and adjacent roads, but such an input method cannot be transferred because of the different number of roads. Finally, the input data of the baseline methods are the historical speed sequence of the target road. All baseline methods are optimized to a best performance.

ARIMA: It is a method based on time series analysis and is a classic statistical model for predicting traffic flow. The number of autoregressive terms, the number of nonseasonal differences needed for stationarity and the number of lagged forecast errors in the prediction equation are set to 5, 2 and 0. That is, using the ARIMA (5, 2, 0) forecasting equation.

CNN: It refers to a CNN-based traffic flow prediction model. This paper uses a single-layer CNN, the size of the convolution kernel is [3, 1], the stride of the convolution kernel is [1, 1], with no padding, and the number of channels is 16.

Finally, a multi-task linear regression model is trained on the top layer.

LSTM: This paper uses a three-layer LSTM layer with 512 hidden units to process the time series. Flatten the output of the LSTM unit at all moments, and finally trains a multi-task linear regression model on the top layer.

SAEs: Stacked autoencoders are configured with three layers of [64, 256, 512] hidden units for pre-training. Finally train a multi-task linear regression model on the top layer.

ResMLP: As one of the most advanced deep learning algorithms [46], this paper simply applies it to traffic speed prediction. The data at each time is regarded as a picture with only one pixel, and the length of the time series is regarded as the number of channels of the picture. The hidden layer unit is set to 256 and the network depth is set to 12.

The results are shown in Table 5. The prediction effect of IRNet is better than that of the baseline method, and it has a good performance in the speed prediction of short prediction horizon. At the same time, LSTM, as a deep learning algorithm for processing time series, has shown better performance in speed prediction tasks with long prediction horizon. It can be seen in Fig. 10 (a) that the points above y=x of IRNet are offset farther, but the offset points are less than other methods. And in Fig. 10 (b), the error of IRNet is slightly smaller than that of other methods in most of the time between 0-80H, but occasionally a larger error occurs, such as near 60H and after 80H.

D. Transfer Learning

Transfer learning is a popular method in the field of deep learning. It can speed up the training process of the model and can effectively reduce the amount of target data required. However, the limitation of transfer learning is that it requires a large amount of raw data to train a pre-trained model suitable for the source task, learning as much as possible about the raw data. Then the new model for the target task is established and trained by transferring knowledge. Therefore, the source task and the target task must be highly correlated, and general knowledge needs to be learned from the source task. The speed prediction of different roads is a highly related task.
Fig. 10. (a) The scatter plot of true values and predicted values for different methods on the test set. (b) The error plot of true values and predicted values for different methods on the test set.

Fig. 11. The overall strategy of transfer learning.

The baseline methods only use the historical data of target road, while the method in this paper reconstructs the road structure. Therefore, the target road does not affect the structure of the model. Invariant model structure is a precondition for model transfer.

In transfer learning, the parameters of the whole model are updated by training on the raw dataset to obtain a pre-trained model. Then, all of the model parameters are fixed. The model is fine-tuned with very little target data, and only the parameters of the final regression layer are updated. Finally, the target dataset is used to verify the prediction effect of the fine-tuned model. The overall strategy is shown in Fig. 11.

For the various baseline methods, the data of the road NO.10244 is used for training. The trained model is saved as a pre-trained model. The road NO.10384 is selected as the unknown road. And the first ten moments of the road NO.10384 are used as the fine-tuning dataset. The results are shown in Table 6. After fine-tuning, IRNet has a significant improvement over other baseline methods. This is because that baseline methods can only transfer temporal features. However, IRNet can transfer the spatiotemporal features at the same time after reconstruction of the intersection, which greatly improves the transfer ability of the model.

In Fig. 12(a), the predicted values of IRNet from 55km/h to 65km/h are very close to the y=x line, but around 70km/h, the predicted values of IRNet are generally low. In Fig. 12(b), it can also be seen that the overall error values of IRNet are more evenly distributed in the positive and negative regions compared to ResMLP and LSTM.

### E. Discussion

The above experiments verify the influence of spatiotemporal information on the prediction ability of the model. The proposed model is compared with the existing road speed prediction models in terms of prediction effect and transfer ability. The conclusion is that the intersection reconstruction model in this paper can reconstruct the intersection to introduce spatial information and strengthen the transfer ability of the model without affecting the prediction effect.

1) Larger sequence length and spatial width can provide more temporal and spatial information. It means that more spatiotemporal features can be captured and better prediction results can be obtained. However, in Experiment B, increasing the space width and sequence length led to the reduction of short-term prediction effect and increased computational pressure, so this paper chooses more conservative parameters.

2) When the road structure is not processed, the change of the target road results in the structural change of

|      | p=1 | p=2 | p=3 | p=4 | p=5 |
|------|-----|-----|-----|-----|-----|
| RMSPE |     |     |     |     |     |
| ARIMA | 3.76% | 7.10% | 8.79% | 9.91% | 11.47% |
| SAEs  | 5.59% | 6.37% | 6.34% | 6.53% | 6.55% |
| CNN   | 14.91% | 14.59% | 17.03% | 16.94% | 13.31% |
| LSTM  | 4.78% | 5.48% | 7.37% | 6.00% | 6.79% |
| ResMLP| 4.65% | 6.30% | 6.53% | 7.31% | 7.54% |
| IRNet | 2.70% | 5.26% | 4.83% | 5.95% | 5.56% |

|      |     |     |     |     |     |
|------|-----|-----|-----|-----|-----|
| MAPE |     |     |     |     |     |
| ARIMA | 2.19% | 4.14% | 5.68% | 6.87% | 8.09% |
| SAEs  | 4.05% | 4.57% | 4.68% | 4.85% | 4.86% |
| CNN   | 8.05% | 9.21% | 10.18% | 10.40% | 9.02% |
| LSTM  | 3.79% | 4.29% | 5.58% | 4.93% | 4.87% |
| ResMLP| 2.87% | 4.05% | 4.54% | 5.38% | 5.72% |
| IRNet | 2.10% | 4.38% | 3.65% | 4.89% | 4.52% |
the adjacent roads. Therefore, if the existing road speed prediction model wants to form a pre-trained model suitable for all target roads, it can only use the temporal information of the target road or model the entire road network to introduce spatial information. The intersection reconstruction method proposed in this paper unifies the adjacent road structure of each target road. It solves the problem that the existing road speed prediction models are difficult to extract spatiotemporal information at the same time. Moreover, in the comparison results of experiment C, it can be seen that the effect of IRNet is slightly ahead of the mainstream time series network LSTM and Spatial-LSTM. Finally, self-attention mechanism is used to combine the spatiotemporal features. Through a series of experiments based on the real dataset PeMS, we prove that IRNet has a better prediction effect than some commonly used baseline models, and has a good level of generalization performance of the model.

In future work, this paper will optimize this work from the following aspects.

First of all, this paper only uses part of the data of one road for training. In theory, after the intersection is reconstructed, the data of multiple roads can be combined into one dataset for training, which will increase the robustness of the model.

Secondly, the intersection reconstruction method in this paper only compares the correlation between multiple roads at the same intersection and then sorts them. Instead of quantifying the correlation between roads to further process the historical data of adjacent roads, which results in partial distortion of spatial information.

Finally, it is possible to delve deeper into the relationship between the spatiotemporal extent and the prediction span. Using the smallest possible spatiotemporal extent to achieve a better prediction effect can not only save computing resources, but also avoid redundant information interference.

VI. Conclusions

This paper propose the IRNet (Transferable Intersection Reconstruction Network) combined with intersection reconstruction method specifically for road speed prediction. After each road is reconstructed into the same structure, the common features of each reconstructed intersection are captured by the convolutional layer, and the temporal correlation and the spatial correlation are obtained through Temporal-LSTM and Spatial-LSTM. Finally, self-attention mechanism is used to combine the spatiotemporal features. Through a series of experiments based on the real dataset PeMS, we prove that IRNet has a better prediction effect than some commonly used baseline models, and has a good level of generalization performance of the model.

In future work, this paper will optimize this work from the following aspects.

First of all, this paper only uses part of the data of one road for training. In theory, after the intersection is reconstructed, the data of multiple roads can be combined into one dataset for training, which will increase the robustness of the model.

Secondly, the intersection reconstruction method in this paper only compares the correlation between multiple roads at the same intersection and then sorts them. Instead of quantifying the correlation between roads to further process the historical data of adjacent roads, which results in partial distortion of spatial information.

Finally, it is possible to delve deeper into the relationship between the spatiotemporal extent and the prediction span. Using the smallest possible spatiotemporal extent to achieve a better prediction effect can not only save computing resources, but also avoid redundant information interference.

REFERENCES

[1] Y. Zhang, Z. Chen, G. Li, Y. Liu, and J. Early, “Integrated velocity prediction method and application in vehicle-environment cooperative control based on internet of vehicles,” IEEE Transactions on Vehicular Technology, vol. PP, no. 99, pp. 1–1, 2021.

[2] Y. Zhang, L. Chu, Y. Ou, C. Guo, and X. Tang, “A cyber-physical system-based velocity-profile prediction method and case study of application in plug-in hybrid electric vehicle,” IEEE Transactions on Cybernetics, vol. PP, no. 99, pp. 1–12, 2019.

[3] Y. Zheng, L. Capra, O. Wolfson, and H. Yang, “Urban computing concepts, methodologies, and applications,” Acm Transactions on Intelligent Systems & Technology, vol. 5, no. 3, pp. 1–55, 2014.

[4] A. M. Nagy and S. Vilmos, “Survey on traffic prediction in smart cities,” Pervasive & Mobile Computing, pp. S1 574 119 217 306 521–, 2018.
[5] L. A. Pipes, “Car following models and the fundamental diagram of road traffic,” Transp.Res., vol. 1, no. 1, pp. 21–29, 1967.
[6] L. Duckstein, E. A. Unwin, and E. T. Boyd, “Variable perception time in car following and its effect on model stability,” IEEE Transactions on Man-Machine Systems, vol. 11, no. 3, pp. 149–156, 1970.
[7] L. Neubert, H. Y. Lee, and M. Schreckenberg, “Density waves and jamming transition in cellular automaton models for traffic flow,” Journal of Physics A General Physics, vol. 32, no. 37, pp. 6517–6525(9), 1998.
[8] K. Rawat, V. K. Katiyar, and P. Gupta, “Two-lane traffic flow simulation model via cellular automaton,” International Journal of Vehicular Technology, 2012(2011-8-22), vol. 2012, pp. 1–6, 2011.
[9] M. Guo, X. Xiao, and J. H. Lan, “A summary of the short-time traffic flow forecasting methods,” Techniques of Automation and Applications, 2009.
[10] Wheelwright and Steven C, Forecasting methods for management. Forecasting methods for management.
[11] J. Guo, H. He, and C. Sun, “Arima-based road gradient and vehicle velocity prediction for hybrid electric vehicle energy management,” IEEE Transactions on Vehicular Technology, vol. 68, no. 6, pp. 5309–5320, 2019.
[12] C. Li and P. Xu, “Application on traffic flow prediction of machine learning in intelligent transportation,” Neural Computing and Applications, vol. 11, no. 2020.
[13] S. A. Angayarkanni, R. Sivakumar, and Y. V. R. Rao, “Hybrid grey forecasting methods for management,” Journal of Intelligent and Artificial Computing, no. 1, 2020.
[14] Y. Liu and W. Hao, “Prediction of road traffic congestion based on random forest,” in International Symposium on Computational Intelligence & Design, 2017.
[15] L. Mccabe, R. Ricci, and P. Concannon, “Simulation studies for an urban traffic corridor,” Highway Traffic Control, 1975.
[16] H. Yu, Z. Wu, S. Wang, Y. Wang, and X. Ma, “Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks,” Sensors, 2017.
[17] L. Cai, Z. Zhang, J. Yang, Y. Yu, and J. Qin, “A noise-immune kalman filter for short-term traffic flow forecasting,” Physica A: Statistical Mechanics and its Applications, p. 122601, 2019.
[18] J. Zhao, Y. Gao, Z. Bai, H. Wang, and S. Lu, “Traffic speed prediction under non-recurrent congestion: Based on lstm method and beidou navigation satellite system data,” IEEE Intelligent Transportation Systems Magazine, vol. 11, no. 2, pp. 70–81, 2019.
[19] T. Wang, A. Hussain, Q. Sun, S. E. Li, and C. Jiuhua, “The prediction of urban road traffic congestion by using a deep stacked long short-term memory network,” IEEE Intelligent Transportation Systems Magazine, vol. 14, no. 4, pp. 102–120, 2022.
[20] X. Ma, D. Zhuang, Z. He, J. Ma, 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, 2017.
[21] X. Cheng, R. Zhang, J. Zhou, and W. Xu, “Deeptransport: Learning spatial-temporal dependency for condition forecasting,” in 2018 International Joint Conference on Neural Networks (IJCNN), 2018, pp. 1–8.
[22] L. Zhao, Y. Song, C. Zhang, Y. Liu, and H. Li, “T-Gen: A temporal graph convolutional network for traffic prediction,” IEEE Transactions on Intelligent Transportation Systems, vol. PP, no. 99, pp. 1–11, 2019.
[23] Y. Shin and Y. Yoon, “Incorporating dynamicity of transportation network with multi-weight traffic graph convolutional network for traffic forecasting,” IEEE Transactions on Intelligent Transportation Systems, vol. 20, no. 99, pp. 1–11, 2020.
[24] M. Lv, Z. Hong, L. Chen, T. Chen, and S. Ji, “Temporal multi-graph convolutional network for traffic flow prediction,” IEEE Transactions on Intelligent Transportation Systems, vol. PP, no. 99, pp. 1–12, 2020.
[25] H. N. Koutsopoulos and H. Farah, “Latent class model for car following behavior,” Transportation Research Part B, vol. 46, no. 5, pp. 563–578, 2012.
[26] X. Liu, J. Xu, M. Li, L. Wei, and H. Ru, “General-logistic-based speed-density relationship model incorporating the effect of heavy vehicles,” Mathematical Problems in Engineering, vol. 2019, no. 1934, pp. 1–10, 2019.
[27] R. Herman, T. Lam, and I. Prigogine, “Kinetic theory of vehicular traffic,” Transportation, vol. 6, no. 4, pp. 440–452, 1972.
[28] S. Xu and L. Yu, “Evaluation of traffic simulation models for supporting its development,” 2000.
[29] D. R. Drew, “Traffic flow theory and control,” McGraw-Hill Series in Transportation, vol. 316, 1968.
Pengyu Fu received the B.S. degree in vehicle engineering from Chongqing University, Chongqing, China, in 2019. He is currently working toward the M.S. degree in vehicle engineering from Jilin University, Changchun, China.

His current research interests include machine learning and neural networks.

Liang Chu was born in 1967. He received the B.S., M.S., and Ph.D. degrees in vehicle engineering from Jilin University, Changchun, China. He is currently a Professor and the Doctoral Supervisor with the College of Automotive Engineering, Jilin University. His research interests include the driving and braking theory and control technology for hybrid electric vehicles, which conclude powertrain and brake energy recovery control theory and technology on electric vehicles and hybrid vehicles, theory and technology of hydraulic antilock braking and stability control for passenger cars, and the theory and technology of air brake ABS, and the stability control for commercial vehicle.

Dr. Chu has been a SAE Member. He was a member at the Teaching Committee of Mechatronics Discipline Committee of China Machinery Industry Education Association in 2006.

Zhuoran Hou received the B.S. degree in vehicle engineering from Chongqing University, Chongqing, China, in 2017. and the M.S. in Automotive Engineering from Jilin University, China, in 2020. He is currently pursuing continuous academic program involving doctoral studies in automotive engineering with Jilin University, Changchun, China.

His research interests include basic machine learning, optimal energy management strategy about plug-in hybrid vehicles.

Jincheng Hu (Student Member, IEEE) received the B.E. degree in information security from the Tianjin University of Technology, Tianjin, China, in 2019 and the M. Sc. degree in information security from the University of Glasgow, Glasgow, UK, in 2022. He is currently working toward the Ph.D. degree in Automotive with the Loughborough university, Loughborough, UK.

His research interests include reinforcement learning, deep learning, cyber security, and energy management.

Yanjun Huang (Member, IEEE) received the Ph.D. degree in mechanical and mechatronics engineering from the University of Waterloo, Waterloo, Canada, in 2016. He is currently a Professor with the School of Automotive Studies, Tongji University, Shanghai, China. He has authored several books and over 50 papers in journals and conferences. His research interests include the vehicle holistic control in terms of safety, energy saving, and intelligence, including vehicle dynamics and control, hybrid electric vehicle/electric vehicle optimization and control, motion planning and control of connected and autonomous vehicles, and human-machine cooperative driving.

Dr. Huang serves as the Associate Editor and Editorial Board Member for the IET Intelligent Transport System, Society of Automotive Engineers (SAE) International Journal of Commercial vehicles, International Journal of Vehicle Information and Communications, Automotive Innovation, etc.

Yuanjian Zhang (Member, IEEE) received the M.S. in Automotive Engineering from the Coventry University, UK, in 2013, and the Ph.D. in Automotive Engineering from Jilin University, China, in 2018. In 2018, he joined the University of Surrey, Guildford, UK, as a Research Fellow in advanced vehicle control. From 2019 to 2021, he worked in Sir William Wright Technology Centre, Queen’s University Belfast, UK.

He is currently a Lecturer with the Department of Aeronautical and Automotive Engineering, Loughborough University, Loughborough, U.K. He has authored several books and more than 50 peer-reviewed journal papers and conference proceedings. His current research interests include advanced control on electric vehicle powertrains, vehicle-environment-driver cooperative control, vehicle dynamic control, and intelligent control for driving assist system.