

# MatVMD-TGCN: A Fusion Model for Traffic Speed Prediction

Duoliang Liu, Meijuan Li, Xuebo Chen\*

**Abstract**—Traffic speed prediction, as an important part of intelligent transportation system, is of great significance for traffic planning, management and control. Traffic speed are spatially and temporally dependent due to the influence of roadway topology, tidal changes in traffic flow, and other factors. Compared to the use of time series prediction models for traffic speed prediction, Temporal Graph Convolutional Network (TGCN) model can capture the spatio-temporal features of traffic speed and achieve better prediction performance. Increasingly complex real-world traffic situations place higher demands on the predictive power of models. Therefore, this paper proposes a MatVMD-TGCN fusion model that combines signal processing method and deep neural network. Where Matrix Variational Mode Decomposition (MatVMD) is used to decompose a multidimensional univariate time series into a number of mode time series. Parallel TGCN learn the patterns and regularities of each mode time series by capturing their spatio-temporal features, which enables the fusion model to have better prediction capability. Use two real traffic speed datasets, SZ-taxi and Los-loop to test fusion model and other baseline models. Comparing with TGCN model under the prediction horizon 3 and 12; For the SZ-taxi datasets, the RMSE of fusion model decrease 44.85% and 25.70% respectively, the accuracy of fusion model increase 17.86% and 10.49% respectively. For the Los-loop datasets, the RMSE of fusion model decrease 47.26% and 46.51% respectively, the accuracy of fusion model increase 4.59% and 6.72% respectively. The experiment results proved that the fusion model not only has better prediction performance, but also has the ability of long and short-term prediction, real-time prediction, and short-term fluctuation prediction.

**Index Terms**—Traffic speed prediction, Variational Mode Decomposition (VMD), Matrix Variational Mode Decomposition (MatVMD), Temporal Graph Convolutional Network (TGCN), MatVMD-TGCN fusion model.

## I. INTRODUCTION

THE rapid growth of the urban population and the rapid development of the automobile industry have put enormous pressure on the transportation system. Traffic congestion, pollution and accidents are increasingly serious problems. In order to improve the traffic efficiency and reduce the energy loss of transportation systems, artificial intelligence technologies have been integrated into transportation services and management, resulting in the development of Intelligent Transportation Systems

(ITS). ITS can anticipate future traffic states by collecting and processing diverse historical data through smart infrastructure and advanced algorithms[1]. The results of the prediction can be used for transportation planning, management and control, thus alleviating problems such as traffic congestion to a certain extent. In traffic states, speed is a fundamental attribute of dynamic traffic, reflecting vehicle motion and traffic efficiency[2]. Advance path planning and traffic intervention based on speed prediction is an effective way to improve transportation efficiency and reduce energy consumption. Therefore, our proposed MatVMD-TGCN fusion model for traffic speed prediction has practical significance.

Existing traffic speed prediction methods are divided into 3 main categories, which are model-driven methods, classical data-driven methods and deep learning model-based methods. Model-driven methods require comprehensive and detailed system modeling based on a priori knowledge[3]. The representative algorithms are “traffic velocity” model[4], cell transmission model[5], queuing theory model[6], etc. Although the model-driven methods is highly interpretable, it is difficult to construct accurate traffic speed prediction models, and the constructed models can not describe traffic speed variations in complex real-world situations. Classical data-driven methods are used to infer trends of change based on statistical patterns of historical data, which are ultimately used to predict and evaluate traffic conditions[1]. Classical data-driven methods can be subdivided into statistical methods and traditional machine learning methods. The Auto Regressive Integrated Moving Average (ARIMA) is one of the dominant methods in statistical methods. ARIMA models traffic speed as a time series and combines auto regressive models, moving average models and integration to capture time dependence[1]. Many variants have been derived from ARIMA, such as SARIMA[7], which integrates seasonal features into prediction, and STARIMA[8], which takes into account finite spatio-temporal features. Compared to simple statistical methods, traditional machine learning methods can deal with high-dimensional data and learn complex patterns of traffic speed dynamics. Common traditional machine learning methods include K-nearest neighbor model[9], Support Vector Regression (SVR) model[10], [11], [12], fuzzy logic model[13], Bayesian network model[14] and neural network model. Deep learning models have received widespread attention for their ability to better capture the dynamic features of traffic speed and achieve the most advanced prediction results available. Some methods based on deep learning models only consider the time dependence of traffic speed, such as using feed-forward neural networks[15], Deep Belief Networks (DBN)[16], Long-Short-Term Memory networks (LSTM) and Gated Recurrent Units (GRUs)[17] to make predictions in the

Manuscript received July 12, 2025, revised September 5, 2025. This research reported herein was supported by the NSFC of China under Grant No. 71571091 and 71771112.

Duoliang Liu is a postgraduate student of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, Liaoning 114051, PR China (e-mail: 18841221708@163.com).

Meijuan Li is a professor of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, Liaoning 114051, PR China (e-mail: 454675273@qq.com).

Xuebo Chen is a professor of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, Liaoning 114051, PR China (\*Corresponding author, e-mail: xuebochen@126.com).

time dimension. In order to take into account the spatial dependence of traffic speed for better prediction, deep learning models such as SAE[18], ST-ResNet[19] have been proposed. Moreover, with the development of convolutional neural networks and graph neural networks, feature architectures combining convolutional neural networks and recurrent neural networks (CNN+RNN)[20], [21], [22], [23], and feature architectures combining graph neural networks and recurrent neural networks (GNN+RNN)[3], [24], [25], [26] were successively proposed. With the emergence of the attention mechanism[27], more and more Transformer-based traffic prediction models have been proposed. Zheng et al proposed GMAN[28] model that employs spatial and temporal attention mechanisms to track dynamic correlations among traffic sensors effectively. Jiang et al proposed PDFormer[29] model which designs a traffic delay-aware feature transformation module that allows explicit modeling of time delays in the propagation of spatial information. While models are pushing the envelope, attention is returning to the data itself. Shao et al proposed STID[30] model which uses the one-hot encoding to integrate the spatial and temporal feature information into input data, thus enhances sample distinguishability. Liu et al proposed STAEformer[31] model which introduces a spatial-temporal adaptive embedding that enhances the capabilities of standard transformers for traffic flow prediction. STID, STAEFormer demonstrated that augmentation of data features by spatio-temporal embedding coding prior to training effectively enhances the richness of data representations, and thus significantly improve the performance of the model. Traffic data can also be considered as sequential signals, and there is no precedent for signal processing of traffic data to enhance the expression of data features. Therefore, we proposed MatVMD-TGCN fusion model which combines the MatVMD signal decomposition method and deep learning model TGCN. Experiments have demonstrated that the fusion model has better predictive performance compared to other models.

Our contributions are as follows:

1. We propose MatVMD for decomposing multidimensional univariate time signals based on the VMD decomposition algorithm.
2. We propose the MatVMD-TGCN fusion model consisting of MatVMD method and deep learning model TGCN.
3. Comparative tests between the fusion model and other baseline models, proves that MatVMD-TGCN fusion model not only has better prediction performance, but also has the ability of long and short-term prediction, real-time prediction, and short-term fluctuation prediction.

The paper is organized as follows: Section II proposes the MatVMD method that can decompose multidimensional univariate time series based on the VMD algorithm. The MatVMD method is combined with the TGCN to obtain the MatVMD-TGCN fusion model, a special class of prediction model that combines signal processing method and deep neural network. Section III evaluates the fusion model using two real traffic speed datasets, SZ-taxi and Los-loop datasets. Compared with other models, the fusion model not only has advanced prediction performance, but also has the capability

of long and short-term prediction, real-time prediction and short-term fluctuation prediction. Section IV concludes that MatVMD-TGCN is suitable for traffic speed prediction based on the experimental results and provides an outlook on the future development of traffic prediction models.

## II. DESCRIPTIONS OF METHODS AND MODELS

### A. Description of the Problem

In this paper, we use the MatVMD-TGCN fusion model to predict traffic speed over a certain time horizon based on historical traffic speed. Since the prediction of traffic speed is done at the spatio-temporal level, the topology of the road network and the historical data of the traffic speed of each road are required. The definition of road network topology graph and the feature matrix are given below:

1) *Definition of Road Network Topology Graph:* We use an unweighted graph  $G = (V, E, A)$  to characterize the topology of the road network by considering a road as a node  $v_i$  and  $V = \{v_1, v_2, \dots, v_n\}$  as the set of road nodes.  $n$  is the number of road nodes,  $E$  is the set of edges and the adjacency matrix  $A$  is used to describe the spatial location relationship between streets.

2) *Definition of Feature Matrix:* Since the fusion model uses a supervised learning mechanism that requires sample data and its corresponding label data. The feature matrix consists of sample feature matrix and label feature matrix which are obtained by preprocessing the historical data in the method given later.

Sample feature matrix  $X \in R^{h \times (n \cdot K)}$ :

$$X = [x_{t-h+1}, x_{t-h+2}, \dots, x_t]^T \quad (1)$$

Label feature matrix  $Y \in R^{p \times n}$ :

$$Y = [y_{t+1}, y_{t+2}, \dots, y_{t+p}]^T \quad (2)$$

In Eq (1), Eq (2) where  $h, p$  are the length of the historical time series and the length of the predicted time series, respectively,  $n$  is the number of road nodes, and  $K$  is the mode decomposition number. Time stamp of sample feature matrix and label feature matrix are  $x_i \in R^{(n \cdot K) \times 1}, y_i \in R^{n \times 1}$ . The road speed prediction problem can be expressed in the following equation:

$$Y = F(G, X) \quad (3)$$

Eq (3) represents the mapping relationship between the sample feature matrix  $X$ , the road network topology graph  $G$  and the label feature matrix  $Y$ ,  $F$  is the nonlinear function that the fusion model tends to fit.

### B. Signal Decomposition Methods

1) *Variational Mode Decomposition (VMD) Algorithm:* Variational Mode Decomposition (VMD)[32] is an effective, adaptive, non-recursive mode decomposition, which aims to decompose the original signal into Intrinsic Mode Function (IMF) components at different frequencies. In other words, VMD can decompose a signal with high complexity, and non-smoothness into multiple, less complexity and relatively smooth sub-signals with different center frequencies. Moreover, it overcomes the endpoint effects and the problem of mode aliasing, such as those existing in the Experience

Mode Decomposition (EMD). In summary, VMD is very suitable for signal decomposition of time series.

The VMD algorithm consists of the following steps: Firstly, the Hilbert transform is applied to each mode to construct the analyzed signal and obtain the one-sided spectrum. Secondly, the spectrum of each IMF is modulated into the corresponding baseband by mixing the exponent with the corresponding estimated center frequency. Finally, the bandwidth of each mode is determined by the Gaussian smoothing of the demodulated signal[33].

The goal of the VMD algorithm is to ensure that the decomposed signal is finite-bandwidth mode component with a center frequency, while minimizing the sum of the estimated bandwidths of each mode under the condition that the sum of all modes is equal to the original signal. This is to maximize the distribution of all frequencies of a mode around its center frequency, thus ensuring the maximum difference between the modes. The resulting constrained variational problem is as follows:

$$\min_{\{u_k\}, \{w_k\}} \left\{ \sum_{k=1}^K \left\| \partial_t \left[ \left( \delta(t) + \frac{j}{\pi t} \right) * u_k(t) \right] e^{-jw_k t} \right\|_2^2 \right\} \quad (4)$$

$$s.t. f(t) = \sum_{k=1}^K u_k(t) \quad (5)$$

In Eq (4), Eq (5)  $f(t)$  is the original signal,  $K$  is the mode decomposition number;  $u_k = \{u_1, u_2, \dots, u_K\}$  is the set of each IMF component,  $w_k = \{w_1, w_2, \dots, w_K\}$  is the set of center frequency of each IMF component.  $\delta(t)$  is the Dirac function;  $*$  denotes the convolution operation in the Hilbert transform. Eq (4) is the objective function to be optimized and Eq (5) is the constraint. The augmented Lagrangian function is introduced to convert the constrained problem into an unconstrained one:

$$L(\{u_k\}, \{w_k\}, \lambda) = \alpha \sum_{k=1}^K \left\| \partial_t \left[ \left( \delta(t) + \frac{j}{\pi t} \right) * u_k(t) \right] e^{-jw_k t} \right\|_2^2 + \left\| f(t) - \sum_{k=1}^K u_k(t) \right\|_2^2 + \left\langle \lambda(t), f(t) - \sum_{k=1}^K u_k(t) \right\rangle \quad (6)$$

In Eq (6)  $\alpha$  denotes the penalty factor, which has a significant impact on data fidelity;  $\lambda$  is Lagrange multiplier. The solution to the original minimization constraint problem is the saddle point of the augmented Lagrangian function in a series of iterative sub-optimizations (Including optimization of each IMF, the center frequency of each IMF and Lagrange multiplier). Using the Alternating Direction Multiplier Method (ADMM) to solve the augmented Lagrangian function, we have the update formula of functional  $\hat{u}_k, \hat{w}_k$  and Lagrange multiplier  $\hat{\lambda}_k$ . The complete VMD algorithm is shown below: Where  $\hat{f}(w), \hat{u}_k^n(w), \hat{\lambda}(w)$  are the Fourier transforms of  $f(t), u_k^n(t), \lambda(t)$  under the  $L^2$  norm, respectively.  $n$  indicates iteration times. Time step of dual ascent  $\tau$  indicates the noise tolerance, When there is strong noise in the signal that should not be included in the decomposition, the Lagrange multiplier should be turned off (set  $\tau$  to 0). Specific proofs and derivations of the VMD iteration formula are given in the paper[32].

---

**Algorithm 1:** VMD algorithm [32]

---

*Initialize*  $\{\hat{u}_k^1\}, \{w_k^1\}, \hat{\lambda}^1, n \leftarrow 0$

**Repeat:**

$n \leftarrow n + 1$

**for**  $k = 1 : K$  **do:**

*Update*  $\hat{u}_k$  *for all*  $w \geq 0$  :

$$\hat{u}_k^{n+1}(w) \leftarrow \frac{\hat{f}(w) - \sum_{i=1}^{k-1} \hat{u}_i^{n+1}(w) - \sum_{i=k+1}^K \hat{u}_i^n(w) + \frac{\hat{\lambda}^n(w)}{2}}{1 + 2\alpha(w - w_k^n)^2}$$

*Update*  $w_k$  :

$$w_k^{n+1} \leftarrow \frac{\int_0^\infty w |\hat{u}_k^{n+1}(w)|^2 dw}{\int_0^\infty |\hat{u}_k^{n+1}(w)|^2 dw}$$

**end for**

*Dual ascent for all*  $w \geq 0$

$$\hat{\lambda}^{n+1}(w) \leftarrow \hat{\lambda}^n(w) + \tau \left( \hat{f}(w) - \sum_{k=1}^K \hat{u}_k^{n+1}(w) \right)$$

**until convergence:**  $\sum_{k=1}^K \frac{\|\hat{u}_k^{n+1} - \hat{u}_k^n\|_2^2}{\|\hat{u}_k^n\|_2^2} < \varepsilon$

---

**2) Matrix Variational Mode Decomposition (MatVMD)**

*Method:* VMD is suitable for signal decomposition of one-dimensional time series, while traffic speed prediction is a multidimensional univariate time series prediction task. Therefore we propose the Matrix Variational Mode Decomposition (MatVMD) for signal decomposition of multidimensional univariate time series. MatVMD is extended by VMD, the decomposition steps of MatVMD are as follows:

1. Splitting the multidimensional univariate time series matrix  $T \in R^{h \times n}$  into  $n$   $t_i \in R^{h \times 1}$ :

$$T = [t_0, t_1, \dots, t_{n-1}] \quad (7)$$

2. Using VMD algorithm to decompose each  $t_i$  in  $T$ ,  $n$  IMF components corresponding to each  $t_i$  are obtained:

$$t_i \xrightarrow{VMD} [I_{t_i}^0, I_{t_i}^1, \dots, I_{t_i}^{K-1}], I_{t_i}^k \in R^{h \times 1} \quad (8)$$

3. The mode  $k$  matrix  $M_k$  is obtained by splicing together the  $k$ th IMF component of each  $t_i$ :

$$M_k = [I_{t_0}^k, I_{t_1}^k, \dots, I_{t_{n-1}}^k], M_k \in R^{h \times n} \quad (9)$$

MatVMD can decompose a multidimensional univariate time series matrix  $T \in R^{m \times n}$  into  $K$  mode time series matrix  $M_k \in R^{m \times n}$ , where the mode decomposition number  $K$  of MatVMD is the same as the mode decomposition number  $K$  of VMD. MatVMD aims to decompose a multidimensional univariate time series with high complexity and insignificant signal features into several mode time series with significantly reduced complexity and prominent signal features, thus facilitating the subsequent deep neural networks to learn their patterns and regularities. In order to provide an intuitive description of the MatVMD method, Decomposition of the historical time series of road traffic speed where sensors numbered 772151, 718371 and 717489 are located in the Los-loop datasets uses the MatVMD method (Mode decomposition number  $K$  is set to 6). The historical traffic speed fluctuation curves are shown in Figure 1.

Each historical time series is decomposed in turn using the VMD algorithm, and the decomposition results are shown in

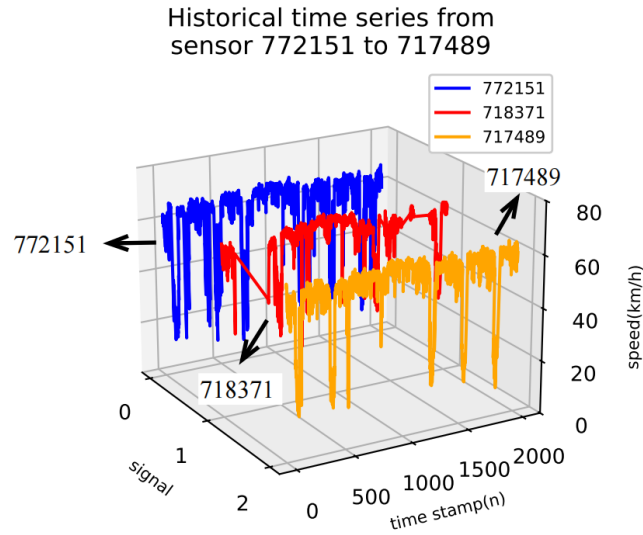


Fig. 1: Multidimensional univariate historical time series of road traffic speed where sensors 772151 to 717489 are located in the Los-loop datasets.

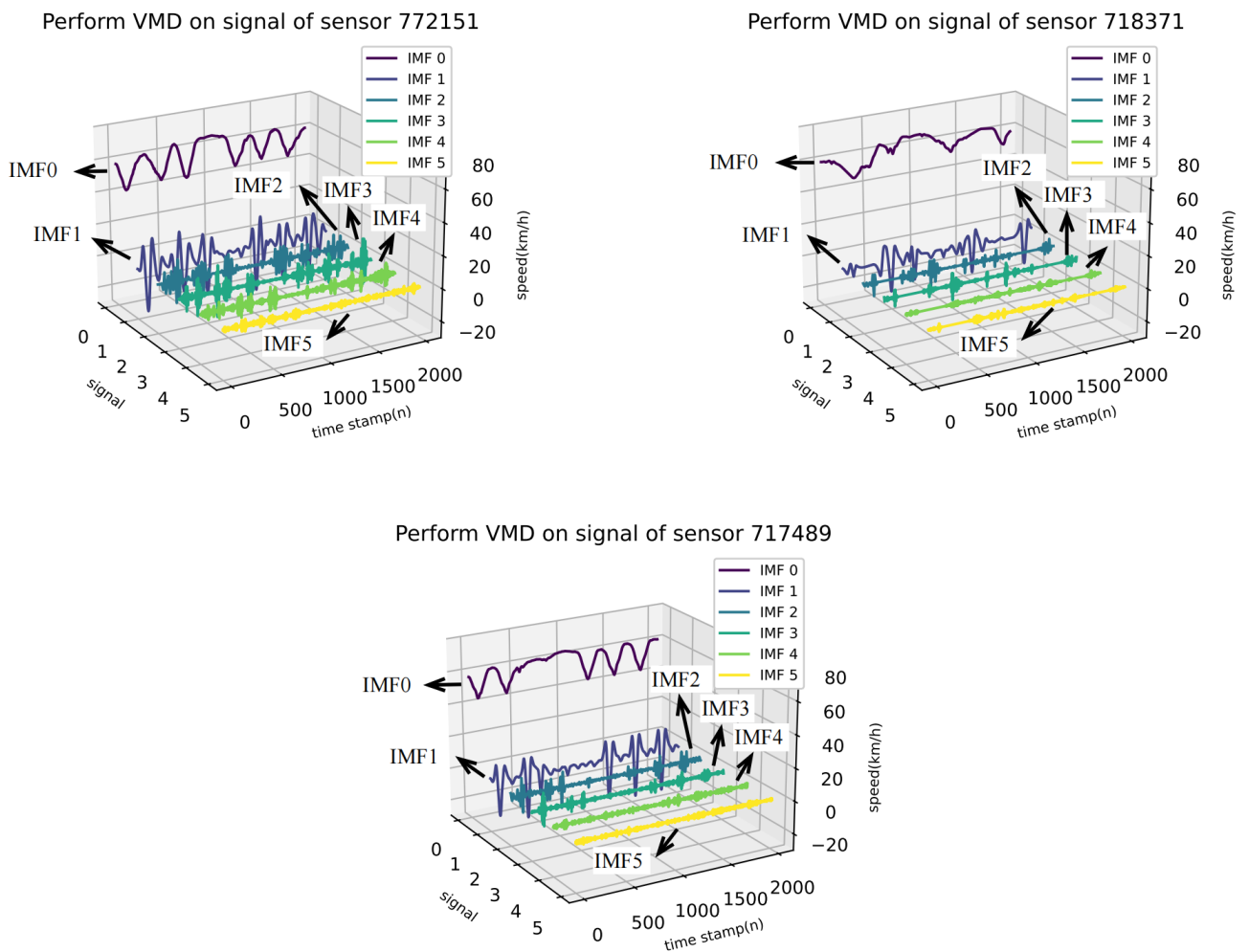


Fig. 2: Decomposition results of VMD for each historical time series in figure 1.

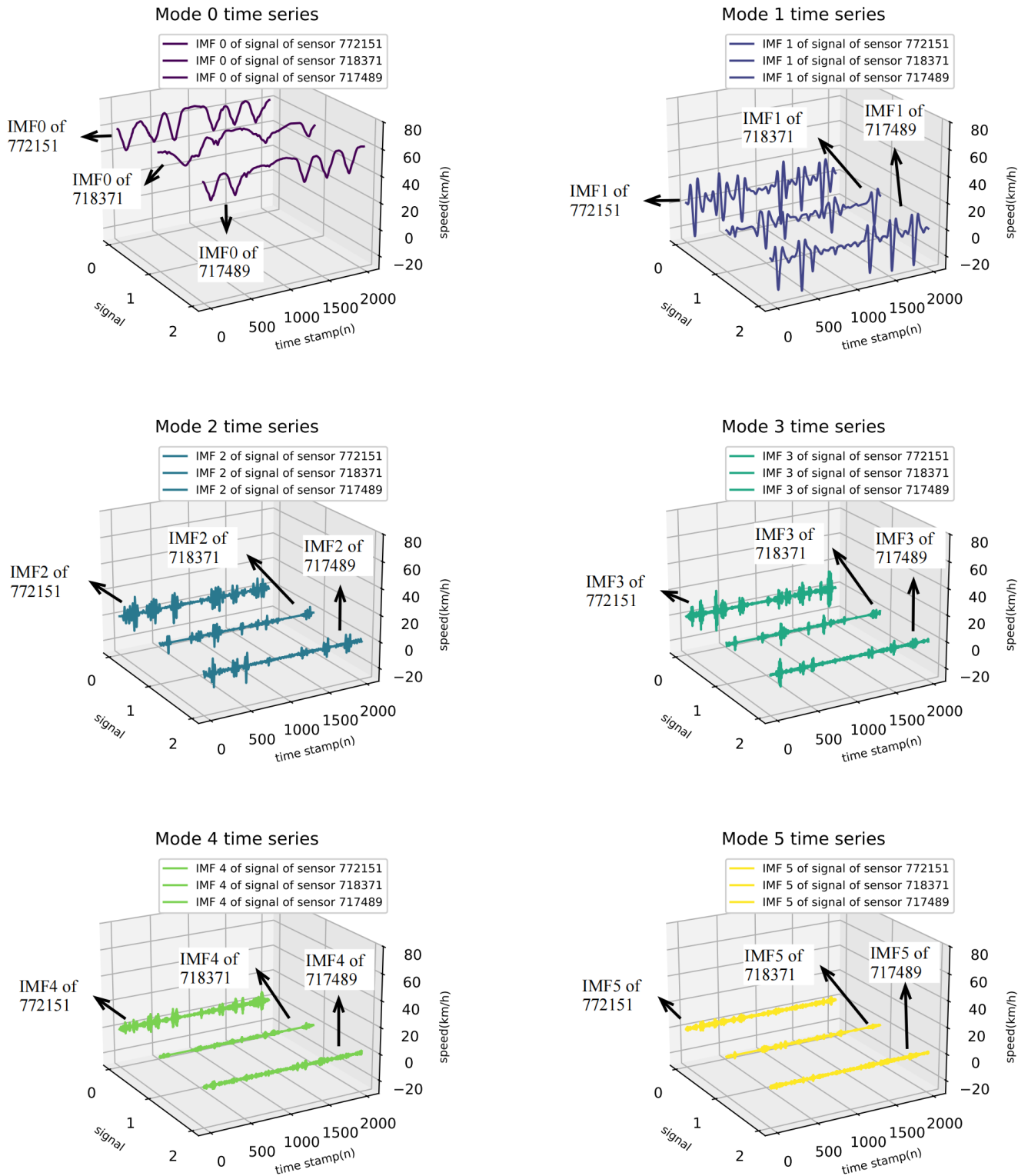


Fig. 3: Each mode time series is obtained by combining the same-level IMFs in figure 2

Figure 2. In Figure 3, mode time series with sequentially increasing frequencies, simple modes, and prominent signal features are obtained by splicing the same-level IMF components from the decomposition results of each historical time series, respectively.

### C. Description of Models

1) *Graph Convolutional Network (GCN)*: GCN[34] is a type of Graph Neural Network (GNN), which is mainly used for processing non-Euclidean spatial data such as graph data. Unlike Euclidean spatial data with regular patterns such as images, text and speech; Graph data represents entities and relationships between entities through nodes and edges. Thus graph data allows a natural, general representation of transportation networks. GCN leverages the ability of the convolutional kernel in Convolutional Neural Network (CNN) to model local structures and the node dependencies prevalent on the graph to become the most active and important branch of GNNs. Expression for the graph convolution operator  $\hat{A} = \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}}$  derived from spectral map theory and Chebyshev polynomials[35].  $\tilde{A}_{n \times n} = A_{n \times n} + I_{n \times n}$  is the self-looped adjacency matrix;  $\tilde{D}$  is the degree matrix of  $\tilde{A}$ ,  $\tilde{D}_{ii} = \sum_{j=1}^n \tilde{A}_{ij}$ ;  $W^{(l)}$  is the trainable parameter matrix for  $l$  layer;  $X$  is the input feature matrix. Suppose  $H^{(l)}, H^{(l+1)}$  is the output of  $l$ th,  $(l+1)$ th layer of the GCN, A multi-layer GCN can be expressed as:

$$H^{(l+1)} = Relu\left(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l)} W^{(l)}\right) \quad (10)$$

2) *Temporal Graph Convolutional Network (TGCN)*: TGCN[3] is improved from GRU[36]. The specific improvement is to construct the  $\hat{A}XW$  structure in GCN by left-multiplying each input of the GRU's reset gate  $r_t$ , update gate  $z_t$  and candidate state  $\tilde{h}_t$  with  $\hat{A} = \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}}$ , thus capturing the spatio-temporal features of the data. A TGCN unit is shown as Figure 4.

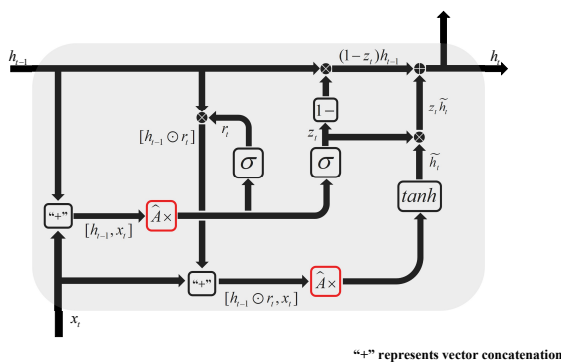


Fig. 4: The structure of TGCN unit

The formulas for the key variables in Figure 4 are as follows:

$$r_t = \text{sigmoid}\left(\hat{A}[h_{t-1}, x_t] \begin{bmatrix} W_{rh} \\ W_{rx} \end{bmatrix} + b_r\right) \quad (11)$$

$$z_t = \text{sigmoid}\left(\hat{A}[h_{t-1}, x_t] \begin{bmatrix} W_{zh} \\ W_{zx} \end{bmatrix} + b_z\right) \quad (12)$$

$$\tilde{h}_t = \tanh\left(\hat{A}[r_t \odot h_{t-1}, x_t] \begin{bmatrix} W_{\tilde{h}h} \\ W_{\tilde{h}x} \end{bmatrix} + b_{\tilde{h}}\right) \quad (13)$$

$$h_t = (1 - z_t) h_{t-1} + z_t \tilde{h}_t \quad (14)$$

Where  $x_t$  is the input of the TGCN unit at the current moment,  $h_{t-1}$  and  $h_t$  are the hidden states of the TGCN unit at the previous and current moments, respectively.  $\tilde{h}_t$  is the candidate hiding state of the TGCN unit at the current moment.  $[W_{rh}^T \ W_{rx}^T]^T$  and  $b_r$  are the weight and bias of reset gate  $r_t$ ;  $[W_{zh}^T \ W_{zx}^T]^T$  and  $b_z$  are the weight and bias of reset gate  $z_t$ ;  $[W_{\tilde{h}h}^T \ W_{\tilde{h}x}^T]^T$  and  $b_{\tilde{h}}$  are the weight and bias of candidate state  $\tilde{h}_t$ ;  $\odot$  represents Hadamard product.

A TGCN consisting of TGCN units is shown in Figure 5; TGCN belongs to recurrent neural network, the input of TGCN are historical time series  $[x_{t-n+1}, x_{t-n+2}, \dots, x_t]$  and graph convolution operator  $\hat{A}$ . The output of TGCN is the last hidden state  $h_t$  of the sequence of hidden states  $[h_{t-n+1}, h_{t-n+2}, \dots, h_t]$ . Experiments proved that the TGCN model is better than the GRU, GCN model in predicting traffic speed.

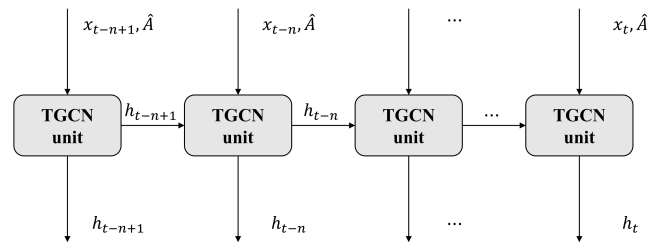


Fig. 5: The structure of TGCN

3) *MatVMD-TGCN Fusion Model*: The MatVMD-TGCN fusion model is a prediction model that combines MatVMD and TGCN. MatVMD is a multidimensional univariate time series decomposition method and TGCN is a deep neural network which is capable of capturing the spatio-temporal features of the data. The fusion model is described in detail below in terms of both data preprocessing and model structure.

a) *Data Preprocessing of MatVMD-TGCN Fusion Model*: The left half of Figure 6 illustrates the data preprocessing process of the fusion model. The specific steps are as follows:

1. The historical multidimensional univariate time series matrix  $h \in R^{T \times N}$  is data normalized and divided into training data  $h_{train} \in R^{(x \cdot T) \times N}$  and validation data  $h_{val} \in R^{((1-x) \cdot T) \times N}$  in the ratio  $x : (1 - x)$ .
2. Perform MatVMD on training data to obtain  $K$  mode time series matrix  $m_K^x = \{M_0^x, M_1^x, \dots, M_{K-1}^x\}$ ,  $M_i^x \in R^{(x \cdot T) \times N}$ ; Perform MatVMD on validation data to obtain  $K$  mode time series matrix  $m_K^{(1-x)} = \{M_0^{(1-x)}, M_1^{(1-x)}, \dots, M_{K-1}^{(1-x)}\}$ ,  $M_i^{(1-x)} \in R^{((1-x) \cdot T) \times N}$ ;

3. The full mode matrix  $M_{\sum}^x \in R^{(x \cdot T) \times (K \cdot N)}$ ,  $M_{\sum}^{(1-x)} \in R^{((1-x) \cdot T) \times (K \cdot N)}$  are obtained by concatenating the  $K$  mode time series matrix  $M_i^x, M_i^{1-x}$ .
4. The training samples  $X_{train} \in R^{B \times L_{in} \times (K \cdot N)}$  and validation samples  $X_{val} \in R^{B \times L_{in} \times (K \cdot N)}$  are obtained using the sliding window method[37] for full mode matrix  $M_{\sum}^x, M_{\sum}^{1-x}$ .
5. The training targets  $Y_{train} \in R^{B \times L_{out} \times N}$  and validation targets  $Y_{val} \in R^{B \times L_{out} \times N}$  are obtained using the sliding window method for both training and validation data.

b) *Structure of MatVMD-TGCN Fusion Model*: The right half of Figure 6 illustrates the structure of the fusion model. The fusion model includes  $K$  channels. The number of channels depends on the mode decomposition number of the MatVMD method. Each channel consists of a TGCN and a fully connected network, and only the mode time series corresponding to that channel are learned and predicted. TGCN learns the pattern and regularity of the mode time series by capturing the spatio-temporal features in the mode time series, and the fully-connected layer is responsible for constructing the mapping relationship between the hidden state output of the TGCN and the channel output. In the forward propagation process of the model, we first perform a matrix decomposition operation on the input matrix to obtain  $K$  mode matrices. Then, we input each mode matrix into the corresponding channel to obtain the prediction matrix of each mode matrix. Finally, we perform matrix summation of the prediction matrix of each mode to obtain the prediction matrix of the input matrix.

#### D. Model Training Methods

The MatVMD-TGCN fusion model is trained using an error back-propagation training method, where the training objective is to minimize the error between the actual traffic value and the predicted value. We use  $V, \hat{V}$  to denote the real traffic speed and the predicted traffic speed, respectively, and the training loss function is shown in Eq (15). The first term is used to describe the error between the real traffic speed and the predicted traffic speed; and the second term is a regularization term for the trainable parameters of the model, which helps to avoid the overfitting problem of the model.  $\lambda$  is the regularization coefficient.

$$Loss = \|V - \hat{V}\| + \lambda L_{reg} \quad (15)$$

### III. EXPERIMENTS

#### A. Description of Experimental Data

The experiments use two real-life datasets from the paper[3], the SZ-taxi datasets and the Los-loop datasets to test the prediction performance of the MatVMD-TGCN fusion model, and the two datasets are described in detail below.

1) *SZ-taxi*: The SZ-taxi datasets records taxi trajectories in Shenzhen from 1 January to 31 January 2015, and 156 main streets in Luohu District of Shenzhen are selected as the main study area. The experimental data consists of two parts: one part is the adjacency matrix  $A \in R^{156 \times 156}$ , and the other part is the historical multidimensional univariate time series matrix  $F \in R^{2976 \times 156}$ .  $A$  is used to describe

the spatial location relationships between streets, where each row contains the interconnections between the corresponding street and other streets: if two streets are connected to each other, then the corresponding position of the adjacency matrix is set to 1, otherwise it is set to 0.  $F$  describes the changes of the traffic speeds over time, and aggregates a total of 2976 pieces of traffic speed data for 156 roads over the range of data recording dates, with a sampling interval of 15 min.

2) *Los-loop*: The Los-loop datasets was collected in real time on highways of Los Angeles County by loop detectors, and traffic speed measured by 207 sensors between 1 March and 7 March 2012 were selected. Similar to the SZ-taxi datasets, the Los-loop datasets consists of two parts: one part is the adjacency matrix  $A \in R^{207 \times 207}$ , and the other part is the historical multidimensional univariate time series matrix  $F \in R^{2016 \times 207}$ .  $F$  aggregates a total of 2016 traffic speed data for 207 roads over the range of data recording dates, with a sampling interval of 5 min. In contrast to the SZ-taxi datasets,  $A$  is calculated based on the distances between sensors in the traffic network and a linear interpolation method is used to fill in missing values in the datasets. The key information of the SZ-taxi and Los-loop datasets is shown in table I

TABLE I: Key Information of the SZ-taxi and Los-loop Datasets

| Datasets | Sensors | Time steps | Sample Interval | Time Range           |
|----------|---------|------------|-----------------|----------------------|
| SZ-taxi  | 156     | 2976       | 15min           | 2015.1.1 - 2015.1.31 |
| Los-loop | 207     | 2016       | 5min            | 2012.3.1 - 2012.3.7  |

#### B. Experimental Data Preprocessing

1) *Perform MatVMD on Original Data*: The parameters of the VMD algorithm in the MatVMD method are set as follows:

1. The balancing parameter of the data-fidelity constraint  $\alpha$ : 2000
2. Time step of dual ascent  $\tanu$ : 0 (No tolerance for noise)
3. Number of mode decompositions  $K$ : 6
4. Keep DC component or not  $DC$ : 0 (Do not keep)
5. Initialization method for each mode, center frequency and Lagrange multiplier  $init$ : 1 (Initialized data follows a uniform distribution)
6. Tolerance of convergence criterion  $tol$ : 1e-7

$\alpha$  is set based on engineering experience; After weighing the prediction performance of the model and the computing power of the experimental equipment, we set the mode decomposition number  $K$  to 6; Therefore, the test model for this experiment is a MatVMD-TGCN fusion model with 6 channels.

2) *Data Normalization*: In order to accelerate the model convergence, improve prediction accuracy and avoid the problem of gradient explosion during model training due to large differences in the scales of features. We use equation (16) to scale the data values to between  $[0, 1]$ . where  $x_i^{norm}$  represents the normalized data and  $\max\{x_i\}$  represents the maximum value in the datasets.

$$x_i^{norm} = \frac{x_i}{\max\{x_i\}} \quad (16)$$

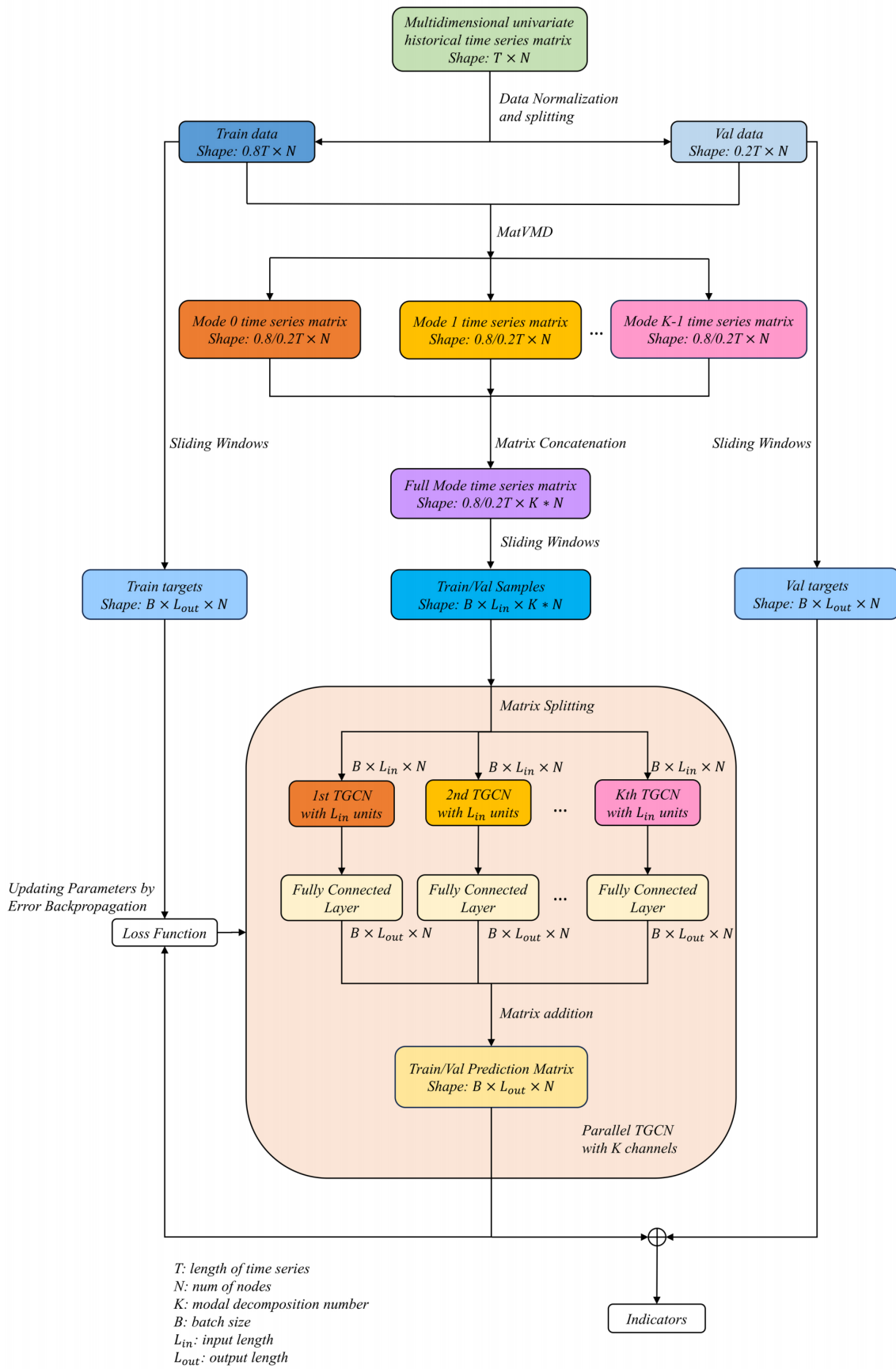


Fig. 6: An overall structure of MatVMD-TGCN fusion model

3) *Training Set and Validation Set Generation*: The prediction task in this experiment is to use the first 12 traffic speed time stamps as sample data for the prediction of next 3, 6, 9, and 12 time stamps. In order to construct a supervised learning task applicable to time series prediction as an objective; The normalized historical traffic speed time series are processed to obtain the training set and validation set according to the method in 2.3.3 in the ratio of 8:2.

### C. Experimental Equipment, Platform and Settings

1) *Experimental Equipment and Platform*: The platform used for the experiment was a DELL laptop with a 4060 discrete graphics card. The deep learning framework Pytorch was chosen as the experimental platform. The version details of torch are as follows:

1. torch = 2.2.2+cu121
2. torch audio = 2.2.2+cu121
3. torch metrics = 1.4.3

2) *Experimental Settings*: In order to ensure the reproducibility of the experiment and to exclude the interference of random factors in the experiment, better demonstrate the excellence of the MatVMD-TGCN model. Before starting the experiment, we make the following settings:

1. seed everything (0)
2. torch.backends.cudnn.deterministic = True
3. torch.backends.cudnn.benchmark = False

### D. Indicators of Model Evaluation

To evaluate the performance of the model, we use the following five indicators to assess the difference between the true road speed  $V$  and the predicted road speed  $\hat{V}$ :

1) *Root Mean Square Error (RMSE)*:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (V_i - \hat{V}_i)^2} \quad (17)$$

2) *Mean Absolute Error (MAE)*:

$$MAE = \frac{1}{n} \sum_{i=1}^n |V_i - \hat{V}_i| \quad (18)$$

3) *Accuracy*:

$$Accuracy = 1 - \frac{\|V - \hat{V}\|_F}{\|V\|_F} \quad (19)$$

4) *Coefficient of Determination ( $R^2$ )*:

$$R^2 = 1 - \frac{\sum_{i=1}^n (V_i - \hat{V}_i)^2}{\sum_{i=1}^n (V_i - \bar{V})^2} \quad (20)$$

5) *Explained Variance Score (var)*:

$$var = 1 - \frac{Var\{V - \hat{V}\}}{Var\{V\}} \quad (21)$$

$RMSE$  and  $MAE$  are the indicators used to measure the prediction error, the smaller the value of both indicates the better the prediction performance of the model. Accuracy is an indicator used to test the predictive accuracy of a model. The larger the value, the higher the predictive accuracy of

the model.  $R^2$  is used to quantify how well the model fits the data and has a value between 0 and 1. The closer the value of  $R^2$  is to 1, the better the fit of the model and the higher the proportion of variation in the dependent variable that can be explained by the independent variable.  $var$  is used to assess how good or bad the model predictions are relative to the actual observations. The closer  $var$  is to 1, the better the model is at explaining changes in the data set and the better the model predicts.

### E. Model Training

1) *Model Hyperparameter Settings*: The hyperparameters associated with the MatVMD-TGCN fusion model include batch size, training epoch, learning rate, regularization factor of the loss function and the hidden state dimension. In this experiment, batch size is set to 64, training epoch is set to 3000. Learning rate is set to 0.001 and regularization factor of the loss function is set to 1.5e-3. The hidden state dimension ( $h_t$ , output of TGCN unit) is set to 100 when the model is trained using the SZ-taxi data and to 64 when the model is trained using the Los-loop datasets. In addition to this, since the Adam optimization algorithm[38] is used during model training, we set the Adam's weight decay rate to 1.5e-3.

2) *Model Trainable Matrices*: Details of trainable matrix in each channel of the fusion model for SZ-taxi and Los-loop datasets as table II and table III.  $r_t$  weight,  $r_t$  bias;  $z_t$  weight,  $z_t$  bias;  $\tilde{h}_t$  weight and  $\tilde{h}_t$  bias are corresponding to the  $[W_{rh}^T, W_{rx}^T]^T$ ,  $b_r$ ;  $[W_{zh}^T, W_{zx}^T]^T$ ,  $b_z$ ;  $[W_{\tilde{h}th}^T, W_{\tilde{h}tx}^T]^T$ ,  $b_{\tilde{h}}$  in Eq(11), (12), (13).

TABLE II: Details of the trainable matrices in each channel of the fusion model for SZ-taxi datasets

| Trainable matrix name             | Shape      | Parameters number |
|-----------------------------------|------------|-------------------|
| $r_t$ weight in TGCN unit         | [101, 100] | 10100             |
| $r_t$ bias in TGCN unit           | [156, 100] | 15600             |
| $z_t$ weight in TGCN unit         | [101, 100] | 10100             |
| $z_t$ bias in TGCN unit           | [156, 100] | 15600             |
| $\tilde{h}_t$ weight in TGCN unit | [101, 100] | 10100             |
| $\tilde{h}_t$ bias in TGCN unit   | [156, 100] | 15600             |
| linear.weight                     | [3, 100]   | 300               |
| linear.bias                       | [3]        | 3                 |

TABLE III: Details of the trainable matrices in each channel of the fusion model for Los-loop datasets

| Trainable matrix name             | Shape     | Parameters number |
|-----------------------------------|-----------|-------------------|
| $r_t$ weight in TGCN unit         | [65, 64]  | 4160              |
| $r_t$ bias in TGCN unit           | [207, 64] | 13248             |
| $z_t$ weight in TGCN unit         | [65, 64]  | 4160              |
| $z_t$ bias in TGCN unit           | [207, 64] | 13248             |
| $\tilde{h}_t$ weight in TGCN unit | [65, 64]  | 4160              |
| $\tilde{h}_t$ bias in TGCN unit   | [207, 64] | 13248             |
| linear.weight                     | [3, 64]   | 192               |
| linear.bias                       | [3]       | 3                 |

3) *Model Training Process*: The complete training flow of the fusion model is shown in Fig 7, where the data preprocessing methods and parameter settings have been described in detail in 3.2. During the training process, after completing a round of training, the model is evaluated using the validation set to obtain the individual indicators of the

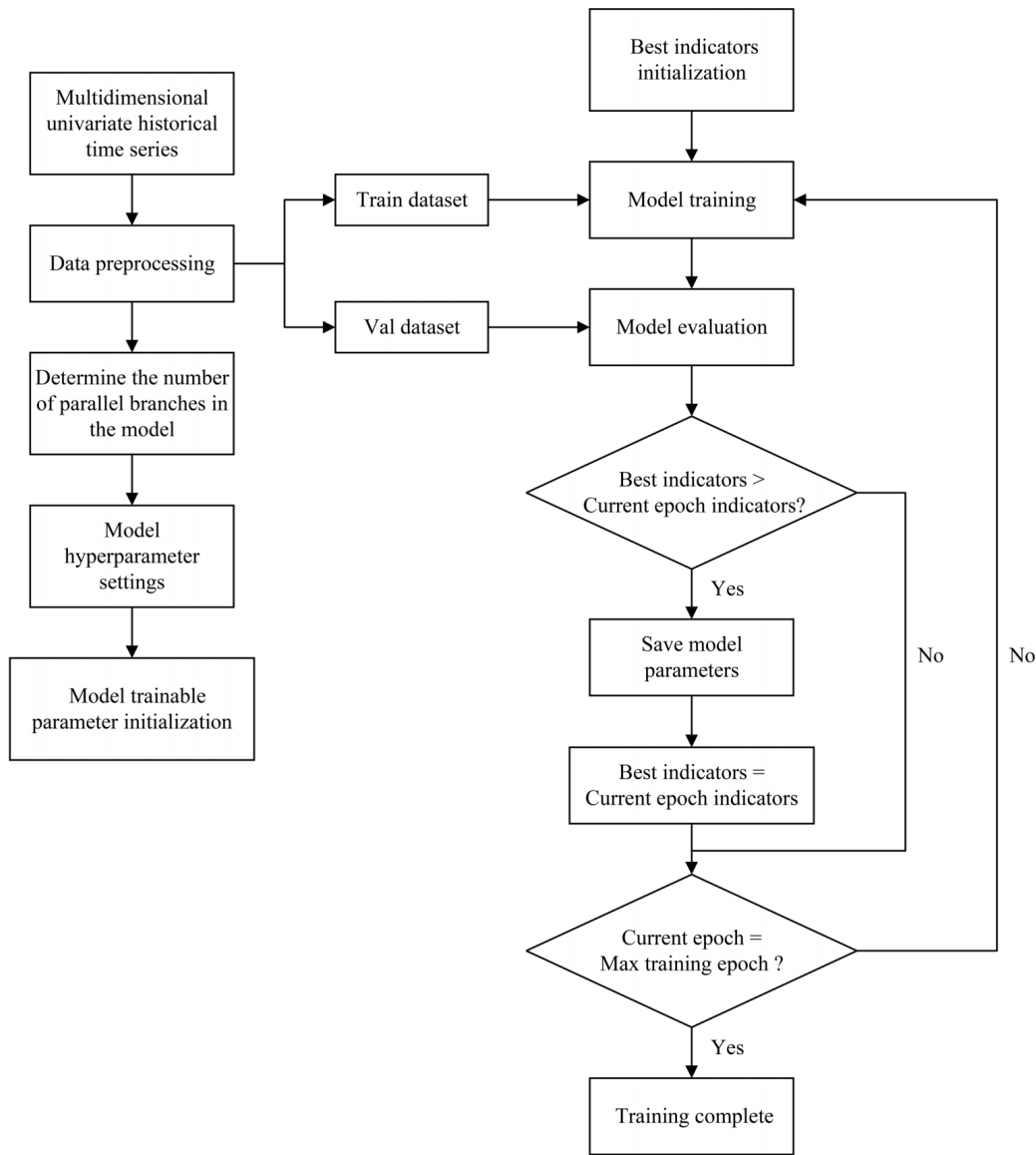


Fig. 7: The complete training process for the MatVMD-TGCN fusion model

model. At the end of the iteration, the best indicators and their corresponding parameters of the optimal model are finally retained.

#### F. Analysis of Experimental Results

We compare the performance of MatVMD-TGCN fusion model with the performance of the following models:

1. History Average model (HA)[39]: Use of average flow information from historical periods as prediction.
2. Auto Regressive Integrated Moving Average (ARIMA).
3. Support Vector Regression model (SVR): Train the model using historical data and obtain the relationship between inputs and outputs. Then trained model is used to predict future traffic data.
4. Convolutional Neural Network (CNN).
5. Graph Convolutional Neural Network (GCN).
6. Gated Recurrent Units (GRU).
7. Transformer consists of Encoder Block with attention mechanism
8. Temporal Graph Convolutional Network (TGCN).

Table IV, V show the predictive indicators of the statistical model HA, ARIMA; machine learning model SVR; deep learning model CNN, GCN, GRU, Transformer, TGCN and MatVMD-TGCN fusion model on SZ-taxi and Los-loop datasets.

Figure 8 shows the prediction results of the deep learning models CNN, GRU, Transformer, TGCN and the fusion model for the speed of the road traffic where sensor No. 92856 is located in the SZ-taxi datasets for different prediction range conditions from 29th January to 31th January. Figure 9 shows the prediction results of the deep learning models CNN, GRU, Transformer, TGCN and the fusion model for the speed of the road traffic where sensor No. 772151 is located in the Los-loop datasets for different prediction range conditions on 7 March. The road on which the sensor is located experienced morning and evening peaks from 7:00 to 10:00 and 15:00 to 18:00 on 7 March, with short periods of traffic congestion at around 15:00 and 16:00 due to uncertainties such as traffic accidents.

As can be seen from Table IV, V and Figure 8,9; the advantages of the MatVMD-TGCN fusion model are:

TABLE IV: Predictive indicators of MatVMD-TGCN and other models on SZ-taxi datasets

| Datasets |                 | SZ-taxi |        |        |        |        |        |             |        |               |
|----------|-----------------|---------|--------|--------|--------|--------|--------|-------------|--------|---------------|
| Horizon  | Indicators      | HA      | ARIMA  | SVR    | CNN    | GCN    | GRU    | Transformer | TGCN   | MatVMD-TGCN   |
| 3        | <i>RMSE</i>     | 4.2951  | 7.2406 | 4.1455 | 4.2144 | 5.6596 | 4.0738 | 4.0518      | 4.0874 | <b>2.2540</b> |
|          | <i>MAE</i>      | 2.7815  | 4.9824 | 2.6233 | 2.8037 | 4.2367 | 2.6815 | 2.6777      | 2.7129 | <b>1.6479</b> |
|          | <i>Accuracy</i> | 0.7008  | 0.4463 | 0.7112 | 0.7063 | 0.6107 | 0.7161 | 0.7177      | 0.7152 | <b>0.8429</b> |
|          | $R^2$           | 0.8307  | *      | 0.8423 | 0.8372 | 0.6654 | 0.8479 | 0.8496      | 0.8469 | <b>0.9534</b> |
|          | <i>var</i>      | 0.8307  | 0.0035 | 0.8424 | 0.8372 | 0.6655 | 0.8479 | 0.8497      | 0.8469 | <b>0.9535</b> |
| 6        | <i>RMSE</i>     | 4.2951  | 6.7899 | 4.1628 | 4.3148 | 5.6918 | 4.1294 | 4.0616      | 4.1197 | <b>2.4434</b> |
|          | <i>MAE</i>      | 2.7815  | 4.6765 | 2.6875 | 2.8917 | 4.2647 | 2.7344 | 2.7071      | 2.7317 | <b>1.7698</b> |
|          | <i>Accuracy</i> | 0.7008  | 0.3845 | 0.7100 | 0.6993 | 0.6085 | 0.7122 | 0.7169      | 0.7129 | <b>0.8297</b> |
|          | $R^2$           | 0.8307  | *      | 0.8410 | 0.8294 | 0.6616 | 0.8438 | 0.8489      | 0.7115 | <b>0.9453</b> |
|          | <i>var</i>      | 0.8307  | 0.0081 | 0.8413 | 0.8294 | 0.6617 | 0.8438 | 0.8494      | 0.8445 | <b>0.9453</b> |
| 9        | <i>RMSE</i>     | 4.2951  | 6.7852 | 4.1885 | 4.4210 | 5.7142 | 4.1633 | 4.0614      | 4.1395 | <b>2.7789</b> |
|          | <i>MAE</i>      | 2.7815  | 4.6734 | 2.7359 | 2.9798 | 4.2844 | 2.7577 | 2.6789      | 2.7441 | <b>1.9589</b> |
|          | <i>Accuracy</i> | 0.7008  | 0.3847 | 0.7082 | 0.6918 | 0.6069 | 0.7098 | 0.7169      | 0.7115 | <b>0.8063</b> |
|          | $R^2$           | 0.8307  | *      | 0.8391 | 0.8209 | 0.6589 | 0.8412 | 0.8489      | 0.8430 | <b>0.9292</b> |
|          | <i>var</i>      | 0.8307  | 0.0087 | 0.8397 | 0.8210 | 0.6590 | 0.8412 | 0.8491      | 0.8431 | <b>0.9292</b> |
| 12       | <i>RMSE</i>     | 4.2951  | 6.7708 | 4.2156 | 4.4894 | 5.7361 | 4.1907 | 4.0670      | 4.1570 | <b>3.0888</b> |
|          | <i>MAE</i>      | 2.7815  | 4.6655 | 2.7751 | 3.0681 | 4.3034 | 2.7965 | 2.6902      | 2.7681 | <b>2.1409</b> |
|          | <i>Accuracy</i> | 0.7008  | 0.3851 | 0.7063 | 0.6870 | 0.6054 | 0.7078 | 0.7165      | 0.7102 | <b>0.7847</b> |
|          | $R^2$           | 0.8307  | *      | 0.8370 | 0.8154 | 0.6564 | 0.8391 | 0.8487      | 0.8417 | <b>0.9126</b> |
|          | <i>var</i>      | 0.8307  | 0.0111 | 0.8379 | 0.8155 | 0.6564 | 0.8393 | 0.8485      | 0.8419 | <b>0.9126</b> |

\* means that the values are small enough to be negligible, indicating that the model's prediction effect is poor.

TABLE V: Predictive indicators of MatVMD-TGCN and other models on Los-loop datasets

| Datasets |                 | Los-loop |         |        |        |        |        |             |        |               |
|----------|-----------------|----------|---------|--------|--------|--------|--------|-------------|--------|---------------|
| Horizon  | Indicators      | HA       | ARIMA   | SVR    | CNN    | GCN    | GRU    | Transformer | TGCN   | MatVMD-TGCN   |
| 3        | <i>RMSE</i>     | 7.4427   | 10.0439 | 6.0084 | 5.5677 | 7.7952 | 5.2419 | 6.8261      | 5.1999 | <b>2.7442</b> |
|          | <i>MAE</i>      | 4.0145   | 7.6832  | 3.7285 | 3.1255 | 5.3525 | 2.9853 | 4.2344      | 3.1479 | <b>1.8993</b> |
|          | <i>Accuracy</i> | 0.8733   | 0.8275  | 0.8977 | 0.9052 | 0.8673 | 0.9108 | 0.8838      | 0.9115 | <b>0.9533</b> |
|          | $R^2$           | 0.7121   | *       | 0.8123 | 0.8389 | 0.6843 | 0.8573 | 0.7590      | 0.8595 | <b>0.9609</b> |
|          | <i>var</i>      | 0.7121   | *       | 0.8146 | 0.8392 | 0.6844 | 0.8577 | 0.7628      | 0.8601 | <b>0.9609</b> |
| 6        | <i>RMSE</i>     | 7.4427   | 9.3450  | 6.9588 | 6.6705 | 8.3353 | 6.2884 | 7.4830      | 6.1820 | <b>3.1658</b> |
|          | <i>MAE</i>      | 4.0145   | 7.6891  | 3.7248 | 3.6766 | 5.6118 | 3.5590 | 4.4867      | 3.6907 | <b>2.1699</b> |
|          | <i>Accuracy</i> | 0.8733   | 0.8275  | 0.8815 | 0.8864 | 0.8581 | 0.8929 | 0.8726      | 0.8947 | <b>0.9461</b> |
|          | $R^2$           | 0.7121   | *       | 0.7492 | 0.7697 | 0.6402 | 0.7953 | 0.7113      | 0.8023 | <b>0.9481</b> |
|          | <i>var</i>      | 0.7121   | *       | 0.7523 | 0.7704 | 0.6404 | 0.7956 | 0.7145      | 0.8031 | <b>0.9481</b> |
| 9        | <i>RMSE</i>     | 7.4427   | 10.0508 | 7.7504 | 7.7315 | 8.8036 | 7.0390 | 7.9282      | 6.8495 | <b>3.6123</b> |
|          | <i>MAE</i>      | 4.0145   | 7.6924  | 4.1288 | 4.3175 | 5.9534 | 4.1480 | 4.6966      | 4.1237 | <b>2.4541</b> |
|          | <i>Accuracy</i> | 0.8733   | 0.8273  | 0.8680 | 0.8683 | 0.8500 | 0.8801 | 0.8649      | 0.8833 | <b>0.9385</b> |
|          | $R^2$           | 0.7121   | *       | 0.6899 | 0.6916 | 0.5999 | 0.7444 | 0.6763      | 0.7579 | <b>0.9327</b> |
|          | <i>var</i>      | 0.7121   | *       | 0.6947 | 0.6919 | 0.6001 | 0.7446 | 0.6779      | 0.7582 | <b>0.9327</b> |
| 12       | <i>RMSE</i>     | 7.4427   | 10.0538 | 8.4388 | 8.4063 | 9.2657 | 7.6878 | 8.2410      | 7.4134 | <b>3.9651</b> |
|          | <i>MAE</i>      | 4.0145   | 7.6952  | 4.5036 | 4.8491 | 6.2892 | 4.4825 | 4.8215      | 4.4950 | <b>2.6784</b> |
|          | <i>Accuracy</i> | 0.8733   | 0.8273  | 0.8562 | 0.8568 | 0.8421 | 0.8690 | 0.8596      | 0.8737 | <b>0.9324</b> |
|          | $R^2$           | 0.7121   | *       | 0.6336 | 0.6365 | 0.5583 | 0.6960 | 0.6511      | 0.7173 | <b>0.9191</b> |
|          | <i>var</i>      | 0.7121   | *       | 0.5593 | 0.6365 | 0.5593 | 0.6960 | 0.6519      | 0.7174 | <b>0.9192</b> |

\* means that the values are small enough to be negligible, indicating that the model's prediction effect is poor.

#### 1. Performance of prediction:

The fusion model has excellent predictors in all horizons. Compared to the HA, SVR, CNN, GRU, Transformer and TGCN model. The prediction error of the fusion model is significantly reduced and the prediction accuracy is substantially improved. On the Los-loop datasets, the accuracy under different prediction ranges reaches more than 93%. The  $R^2$ , *var* values of the fusion model under different prediction ranges on two datasets are all greater than 0.90, which indicates that the model has a high degree of fit to the data and strong explanatory power; it can fully learn the patterns and regularities in the historical time series.

#### 2. Ability of long and short-term prediction:

When making short-term prediction (horizon 3, 6), without exception, the fusion model are all significantly ahead in their indicators on both datasets. When making long-term prediction (horizon 9, 12) on SZ-taxi datasets,

the CNN model was even worse than the HA model for each indicator, and the SVR model was comparable to the HA model for each indicator. When making long-term prediction (horizon 9, 12) on Los-loop datasets, the SVR, CNN, and Transformer model are worse than HA model. When the prediction horizon is 12 on SZ-taxi datasets, compared to the most advanced Transformer model, the *RMSE*, *MAE* of the fusion model decreased by 24.05% and 20.42%, respectively; and *Accuracy*,  $R^2$ , *var* improved by 9.52%, 7.53% and 7.55%, respectively. When the prediction horizon is 12 on Los-loop datasets, compared to the most advanced TGCN model, the *RMSE*, *MAE* of the fusion model decreased by 46.51%, 40.41%; and *Accuracy*,  $R^2$ , *var* improved by 6.72%, 28.13% and 28.13% respectively.

#### 3. Ability of real-time prediction:

In different prediction horizon on Los-loop datasets, the prediction results of the CNN, GRU, Transformer and

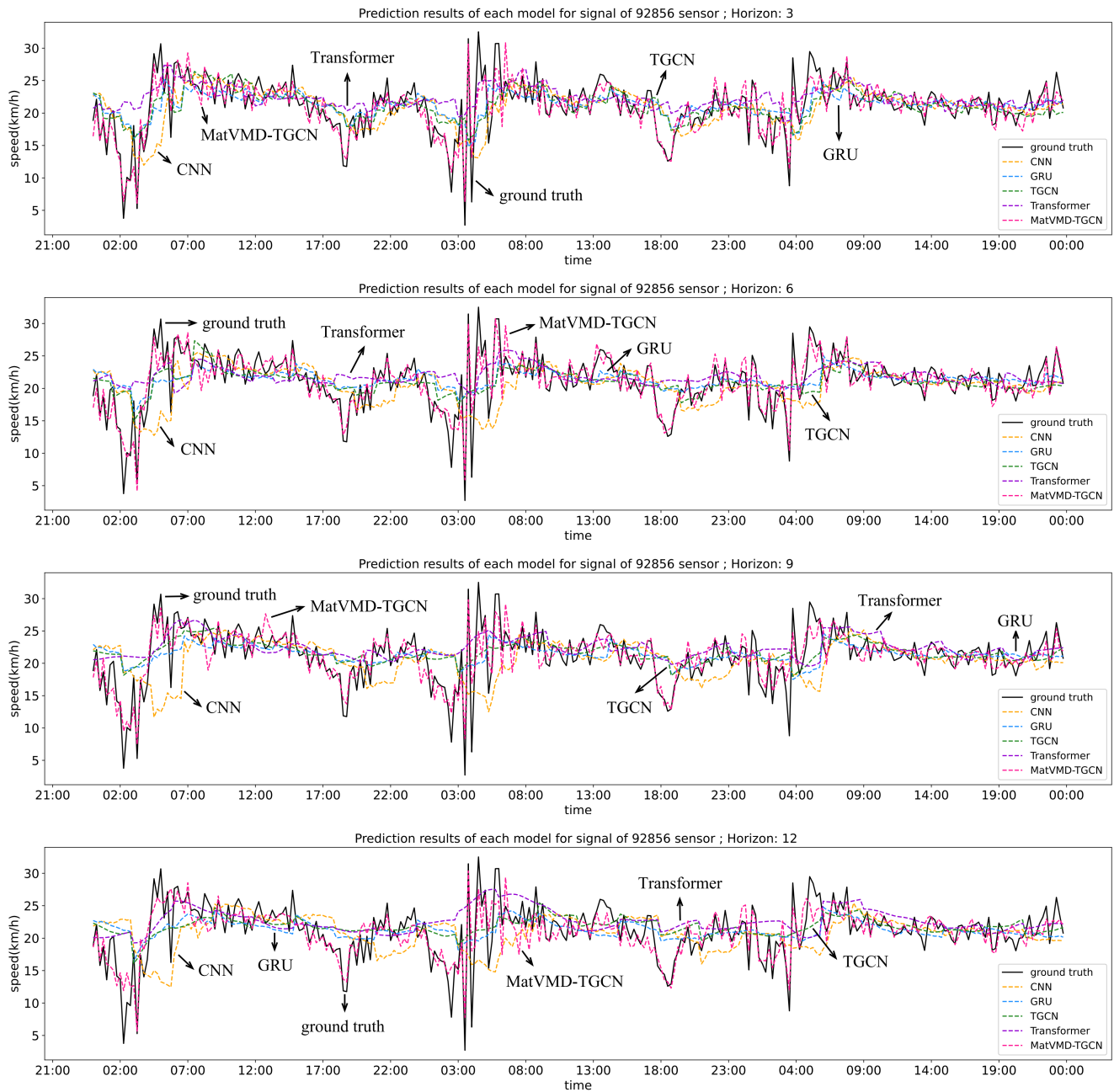


Fig. 8: Comparison of prediction results of each deep learning model with different prediction horizon on SZ datasets

TGCN models have hysteresis compared with the real traffic speed curves; and this hysteresis increases as the prediction horizon increases. The fusion model, on the other hand, does not have hysteresis and is able to make real-time prediction.

#### 4. Ability of short-term fluctuation prediction:

The CNN, GRU, Transformer and TGCN models are not capable of predicting short-term changes in traffic speed, and this disadvantage is highlighted as the prediction horizon increases. For example, On SZ-taxi datasets, CNN, GRU, Transformer and TGCN can not show fluctuations in traffic speed; On Los-loop datasets, when the prediction horizon is 9 and 12, CNN, GRU, Transformer and TGCN models can not reflect the traffic speed fluctuations during the morning and evening peaks well, and can only give the general trend

of their changes; and each of them can not reflect the short-term traffic congestion. Nevertheless, the fusion model has a very satisfactory prediction of the highly nonlinear short-term fluctuations of traffic speed under different prediction horizons.

#### G. Experiments on the mode decomposition $K$ value

The number of channel in the fusion model is determined by the mode decomposition value  $K$ , which plays a key influence on the structure and prediction performance of the model. Therefore, we evaluated the prediction performance of the fusion model with different value of  $K$  on two datasets under prediction horizon 3. The results are shown in Table VI. As shown in Figure 10 and 11, For the Los-loop datasets, The prediction indicators  $RMSE$  of the fusion model decreases and then increases as  $K$  increases. The

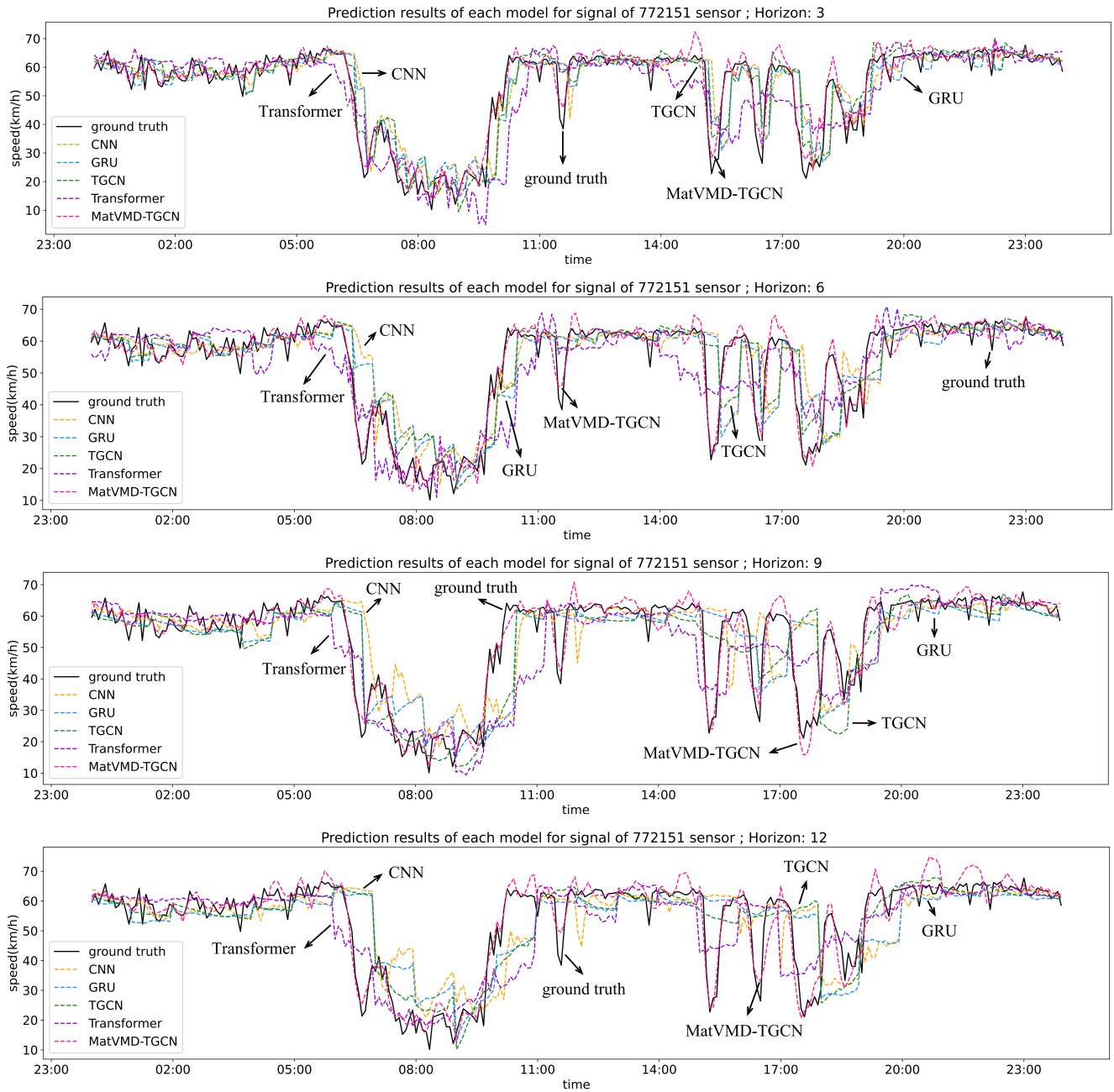


Fig. 9: Comparison of prediction results of each deep learning model with different prediction horizon on Los-loop datasets

TABLE VI: Indicators of MatVMD-TGCN on Los-loop and SZ-taxi datasets with different  $K$ -value under prediction horizon 3

| Datasets  |            | SZ-taxi |        |            |        |        | Los-loop |        |            |        |        |
|-----------|------------|---------|--------|------------|--------|--------|----------|--------|------------|--------|--------|
| $K$ value | Indicators | $RMSE$  | $MAE$  | $Accuracy$ | $R^2$  | $Var$  | $RMSE$   | $MAE$  | $Accuracy$ | $R^2$  | $Var$  |
|           |            |         |        |            |        |        |          |        |            |        |        |
| $K = 4$   |            | 2.9066  | 2.0707 | 0.7975     | 0.9226 | 0.9226 | 3.4403   | 2.3447 | 0.9414     | 0.9385 | 0.9385 |
| $K = 5$   |            | 2.6185  | 1.8853 | 0.8175     | 0.9372 | 0.9372 | 3.2117   | 2.2009 | 0.9453     | 0.9464 | 0.9464 |
| $K = 6$   |            | 2.2540  | 1.6479 | 0.8429     | 0.9534 | 0.9535 | 2.7442   | 1.8993 | 0.9533     | 0.9609 | 0.9609 |
| $K = 7$   |            | 2.3343  | 1.7106 | 0.8373     | 0.9501 | 0.9501 | 2.9295   | 2.0627 | 0.9501     | 0.9554 | 0.9554 |
| $K = 8$   |            | 2.1969  | 1.6163 | 0.8469     | 0.9558 | 0.9558 | 2.9574   | 2.0771 | 0.9497     | 0.9545 | 0.9545 |

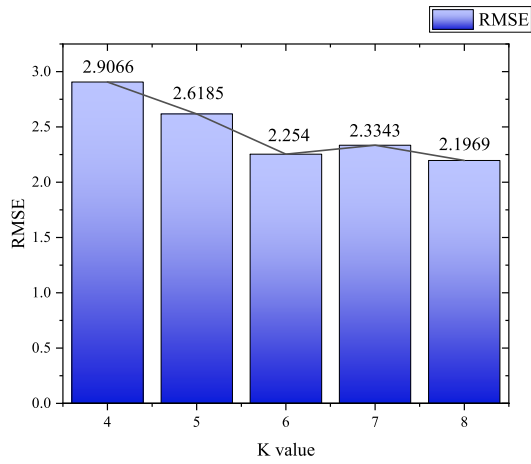


Fig. 10: RMSE of MatVMD-TGCN fusion model with different K value on SZ-taxi datasets

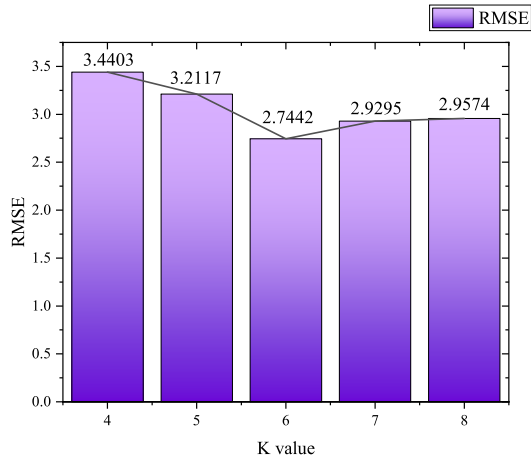


Fig. 11: RMSE of MatVMD-TGCN fusion model with different K value on Los-loop datasets

optimal indicators of fusion model are obtained when  $K$  is chosen as 6. For the SZ-taxi datasets, The optimal indicators of fusion model are obtained when  $K$  is chosen as 8 which is not improved a lot comparing the  $K$  is chosen as 6. Considering the model complexity and computational effort, We set the mode decomposition value as 6 in the experiment.

#### H. Ablation Study

Compared to other model improvement paths by stacking functional layers vertically, MatVMD-TGCN alternatively adopts a horizontal channel stacking strategy to achieve model optimization. Therefore, the objective of ablation experiments is to verify the effectiveness of different channel. We made an ablation study on two datasets under prediction horizon 3. The specific experimental procedure is to increase the number of channels of the fusion model one by one while keeping channel 0 and channel 1, and to evaluate the prediction effect of the fusion model. The results are shown in table VII. For SZ-taxi, The absence of channels has a large impact on the predictive performance of the model; For Los-loop datasets, The prediction performance of the fusion

model has a tendency to enhance as the number of channels increases. It is concluded that each channel is indispensable for the fusion model.

#### I. Robustness Study

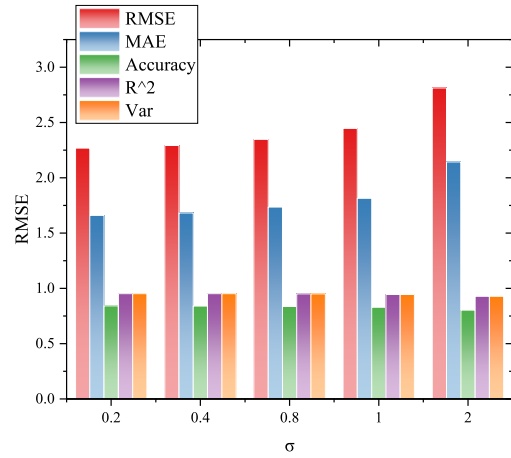


Fig. 12: Robustness study of MatVMD-TGCN fusion model on SZ-taxi datasets under prediction horizon 3 with different standard deviation

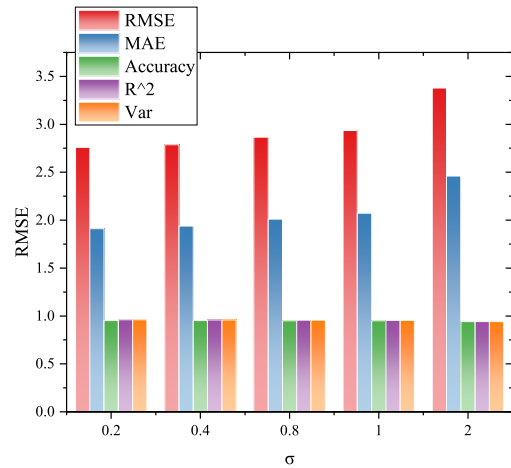


Fig. 13: Robustness study of MatVMD-TGCN fusion model on Los-loop datasets under prediction horizon 3 with different standard deviation

Noise is inevitable in the actual data collection process. To test the noise resistance of the fusion model, we evaluated the robustness of the model through perturbation analysis experiments. We added common random noise—Gaussian noise—to the data during the experiment. Gaussian noise follows a normal distribution  $N \in (0, \sigma^2)$ , where  $\sigma \in (0.2, 0.4, 0.8, 1, 2)$ . The noisy matrix is normalized between 0 to 1 before adding to the original data. Table VIII and figure 12, 13 show the results of robustness study of fusion model on SZ-taxi and Los-loop datasets. It can be seen that noise does not have a significant impact on the predictive performance of the model and is acceptable to a certain extent. Therefore, the fusion model is robust and is able to handle noise issues.

TABLE VII: Ablation study of MatVMD-TGCN fusion model on SZ-taxi and Los-loop datasets under prediction horizon

3

| Datasets          | SZ-taxi     |            |                 |         |            | Los-loop    |            |                 |        |            |
|-------------------|-------------|------------|-----------------|---------|------------|-------------|------------|-----------------|--------|------------|
| Indicators        | <i>RMSE</i> | <i>MAE</i> | <i>Accuracy</i> | $R^2$   | <i>Var</i> | <i>RMSE</i> | <i>MAE</i> | <i>Accuracy</i> | $R^2$  | <i>Var</i> |
| Reserved channels |             |            |                 |         |            |             |            |                 |        |            |
| Channel 0,1       | 10.2129     | 7.9831     | 0.2883          | 0.1615  | 0.1842     | 12.2488     | 10.3178    | 0.7915          | 0.4658 | 0.6800     |
| Channel 0,1,2     | 12.8181     | 9.6647     | 0.1068          | -0.4238 | -0.4482    | 10.2302     | 8.3782     | 0.8258          | 0.5622 | 0.6986     |
| Channel 0,1,2,3   | 13.7109     | 10.2576    | 0.0446          | -0.5448 | -0.4624    | 6.9720      | 5.3844     | 0.8813          | 0.7578 | 0.7906     |
| Channel 0,1,2,3,4 | 12.4325     | 10.2543    | 0.1336          | -0.3754 | -0.3866    | 8.1664      | 6.6218     | 0.8610          | 0.6591 | 0.6703     |
| All Channels      | 2.2540      | 1.6479     | 0.8429          | 0.9534  | 0.9535     | 2.7442      | 1.8993     | 0.9533          | 0.9609 | 0.9609     |

TABLE VIII: Robustness study of MatVMD-TGCN fusion model on SZ-taxi datasets under prediction horizon 3 with different standard deviation

| Datasets   | SZ-taxi     |            |                 |        |            | Los-loop    |            |                 |        |            |
|------------|-------------|------------|-----------------|--------|------------|-------------|------------|-----------------|--------|------------|
| Indicators | <i>RMSE</i> | <i>MAE</i> | <i>Accuracy</i> | $R^2$  | <i>Var</i> | <i>RMSE</i> | <i>MAE</i> | <i>Accuracy</i> | $R^2$  | <i>Var</i> |
| $\sigma$   |             |            |                 |        |            |             |            |                 |        |            |
| 0.2        | 2.2682      | 1.6593     | 0.8419          | 0.9529 | 0.9529     | 2.7589      | 1.9112     | 0.9530          | 0.9604 | 0.9605     |
| 0.4        | 2.2914      | 1.6812     | 0.8403          | 0.9519 | 0.9519     | 2.7872      | 1.9394     | 0.9526          | 0.9596 | 0.9597     |
| 0.8        | 2.3475      | 1.7356     | 0.8364          | 0.9495 | 0.9495     | 2.8658      | 2.0102     | 0.9512          | 0.9573 | 0.9574     |
| 1          | 2.4469      | 1.8136     | 0.8295          | 0.9451 | 0.9451     | 2.9347      | 2.0725     | 0.9500          | 0.9552 | 0.9553     |
| 2          | 2.8127      | 2.1443     | 0.8040          | 0.9275 | 0.9275     | 3.3781      | 2.4596     | 0.9425          | 0.9407 | 0.9407     |

### J. Model Interpretation

Traffic historical time series have the characteristics of high complexity and inconspicuous signal features, which makes it difficult for even deep neural networks to fully learn their patterns and regularities. The traffic historical time series is decomposed into a number of mode time series with significantly reduced complexity and distinct signal features by the MatVMD method; it serves to simplify the complexity and enhance the signal features, thus improving the learning efficiency of deep neural networks. The graph convolution structure in the TGCN unit extracts the spatial features between the same-level IMF components of each sensor signal, and the gating structure extracts the temporal features of the IMF components. Compared to using a single TGCN to learn complex patterns of traffic historical time series, it is more efficient to use parallel TGCNs to learn simple patterns of each mode time series. This is the reason why MatVMD-TGCN achieves excellent prediction results.

### IV. CONCLUSION

For the traffic speed prediction problem, this paper proposes the MatVMD-TGCN fusion model. MatVMD method is used to decompose a multidimensional univariate time series into multiple mode time series; Parallel TGCN learn the patterns and regularities of each mode time series by capturing the spatio-temporal features in each mode time series, which enables the fusion model to have better prediction capability. Comparative tests between the fusion model and other models using two real traffic speed datasets, Los-loop and SZ-taxi, demonstrate that the fusion model is not only advanced in prediction performance, but also has the ability of long and short-term prediction, real-time prediction, and short-term fluctuation prediction. In addition, Robustness study demonstrates that the fusion model is able to handle noise in a certain extent. In summary, the MatVMD-TGCN fusion model can perform the traffic prediction task well.

The hidden unit dimension of the fusion model is the optimal solution determined by a large number of

experiments, and can not be determined by mathematical proofs and logical reasoning. Therefore, the fusion model has the same “black box” property as other deep neural networks. The increasingly complex real-world traffic situation not only poses a greater challenge to the model’s prediction performance, but also puts forward higher requirements on the model’s interpretability. The Deep Stochastic Configuration Network (DSCN)[40] model is a possible direction for the development of future traffic speed prediction models because of its rigorous mathematical theory and scientific construction method, which has strong model interpretability and excellent prediction ability.

### REFERENCES

- [1] Z. Zhou, Z. Yang, Y. Zhang, Y. Huang, H. Chen, and Z. Yu, “A comprehensive study of speed prediction in transportation system: From vehicle to traffic,” *iScience*, vol. 25, no. 3, p. 103909, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2589004222001791>
- [2] E. L. Manibardo, I. Laña, and J. D. Ser, “Deep learning for road traffic forecasting: Does it make a difference?” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 6164–6188, 2022.
- [3] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li, “T-gcn: A temporal graph convolutional network for traffic prediction,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 9, pp. 3848–3858, 2020.
- [4] Q. Wang, L. Li, J. Hu, and B. Zou, “Traffic velocity distributions for different spacings,” *Journal of Tsinghua University. Science and Technology*, vol. 51, 03 2011.
- [5] P. Wei, Y. Cao, and D. Sun, “Total unimodularity and decomposition method for large-scale air traffic cell transmission model,” *Transportation Research Part B: Methodological*, vol. 53, pp. 1–16, 2013. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0191261513000441>
- [6] X. yue Xu, J. Liu, H. ying Li, and J.-Q. Hu, “Analysis of subway station capacity with the use of queueing theory,” *Transportation Research Part C: Emerging Technologies*, vol. 38, pp. 28–43, 2014. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0968090X13002258>
- [7] B. M. Williams and L. A. Hoel, “Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results,” *Journal of Transportation Engineering*, vol. 129, no. 6, pp. 664–672, 2003.
- [8] W. Min and L. Wynter, “Real-time road traffic prediction with spatio-temporal correlations,” *Transportation Research Part C: Emerging Technologies*, vol. 19,

- no. 4, pp. 606–616, 2011. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0968090X10001592>
- [9] Z. Xiao-Li, H. E. Guo-Guang, and L. U. Hua-Pu, “Short-term traffic flow forecasting based on k-nearest neighbors non-parametric regression,” *Journal of Systems Engineering*, vol. 24, no. 2, pp. 178–183, 2009.
  - [10] A. J. Smola† and B. S. Ikonf, “A tutorial on support vector regression,” *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, 2004.
  - [11] C.-H. Wu, J.-M. Ho, and D. Lee, “Travel-time prediction with support vector regression,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 5, no. 4, pp. 276–281, 2004.
  - [12] Z. Yao, C.-F. Shao, and Y.-L. Gao, “Research on methods of short-term traffic forecasting based on support vector regression,” vol. 30, pp. 19–22, 06 2006.
  - [13] H. Yin, S. Wong, J. Xu, and C. Wong, “Urban traffic flow prediction using a fuzzy-neural approach,” *Transportation Research Part C: Emerging Technologies*, vol. 10, no. 2, pp. 85–98, 2002. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0968090X01000043>
  - [14] S. Sun, C. Zhang, and G. Yu, “A bayesian network approach to traffic flow forecasting,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 1, pp. 124–132, 2006.
  - [15] D. Park and L. Rilett, “Forecasting freeway link travel times with a multilayer feedforward neural network,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 14, pp. 357 – 367, 12 2002.
  - [16] W. Huang, G. Song, H. Hong, and K. Xie, “Deep architecture for traffic flow prediction: Deep belief networks with multitask learning,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 2191–2201, 2014.
  - [17] R. Fu, Z. Zhang, and L. Li, “Using lstm and gru neural network methods for traffic flow prediction,” in *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, 2016, pp. 324–328.
  - [18] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, “Traffic flow prediction with big data: A deep learning approach,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2015.
  - [19] J. Zhang, Y. Zheng, and D. Qi, “Deep spatio-temporal residual networks for citywide crowd flows prediction,” 2017. [Online]. Available: <https://arxiv.org/abs/1610.00081>
  - [20] Y. Wu and H. Tan, “Short-term traffic flow forecasting with spatial-temporal correlation in a hybrid deep learning framework,” 2016. [Online]. Available: <https://arxiv.org/abs/1612.01022>
  - [21] X. Cao, Y. Zhong, Y. Zhou, J. Wang, C. Zhu, and W. Zhang, “Interactive temporal recurrent convolution network for traffic prediction in data centers,” *IEEE Access*, vol. 6, pp. 5276–5289, 2018.
  - [22] J. Ke, H. Zheng, H. Yang, and X. M. Chen, “Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach,” *Transportation Research Part C: Emerging Technologies*, vol. 85, pp. 591–608, 2017. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0968090X17302899>
  - [23] H. Yu, Z. Wu, S. Wang, Y. Wang, and X. Ma, “Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks,” *Sensors*, vol. 17, no. 7, 2017. [Online]. Available: <https://www.mdpi.com/1424-8220/17/7/1501>
  - [24] Y. Li, R. Yu, C. Shahabi, and Y. Liu, “Diffusion convolutional recurrent neural network: Data-driven traffic forecasting,” 2018. [Online]. Available: <https://arxiv.org/abs/1707.01926>
  - [25] X. Ren and S. Yuan, “Gcn-lstm combined model for urban link mean speed prediction in the regional traffic network,” in *2022 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCCom/CyberSciTech)*, 2022, pp. 1–7.
  - [26] J. Bai, J. Zhu, Y. Song, L. Zhao, Z. Hou, R. Du, and H. Li, “A3t-gcn: Attention temporal graph convolutional network for traffic forecasting,” *ISPRS International Journal of Geo-Information*, vol. 10, no. 7, 2021. [Online]. Available: <https://www.mdpi.com/2220-9964/10/7/485>
  - [27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” 2023. [Online]. Available: <https://arxiv.org/abs/1706.03762>
  - [28] C. Zheng, X. Fan, C. Wang, and J. Qi, “Gman: A graph multi-attention network for traffic prediction,” 2019. [Online]. Available: <https://arxiv.org/abs/1911.08415>
  - [29] J. Jiang, C. Han, W. X. Zhao, and J. Wang, “Pdformer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction,” 2024. [Online]. Available: <https://arxiv.org/abs/2301.07945>
  - [30] Z. Shao, Z. Zhang, F. Wang, W. Wei, and Y. Xu, “Spatial-temporal identity: A simple yet effective baseline for multivariate time series forecasting,” 2022. [Online]. Available: <https://arxiv.org/abs/2208.05233>
  - [31] H. Liu, Z. Dong, R. Jiang, J. Deng, J. Deng, Q. Chen, and X. Song, “Staformer: Spatio-temporal adaptive embedding makes vanilla transformer sota for traffic forecasting,” 2023. [Online]. Available: <https://arxiv.org/abs/2308.10425>
  - [32] K. Dragomiretskiy and D. Zosso, “Variational mode decomposition,” *IEEE Transactions on Signal Processing*, vol. 62, no. 3, pp. 531–544, 2014.
  - [33] X. Zhang, K. Yang, Q. Lu, J. Wu, L. Yu, and Y. Lin, “Predicting carbon futures prices based on a new hybrid machine learning: Comparative study of carbon prices in different periods,” *Journal of Environmental Management*, vol. 346, p. 118962, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0301479723017504>
  - [34] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” 2017. [Online]. Available: <https://arxiv.org/abs/1609.02907>
  - [35] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, “A comprehensive survey on graph neural networks,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, p. 4–24, Jan. 2021. [Online]. Available: <http://dx.doi.org/10.1109/TNNLS.2020.2978386>
  - [36] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using rnn encoder-decoder for statistical machine translation,” 2014. [Online]. Available: <https://arxiv.org/abs/1406.1078>
  - [37] L. Dong, D. Fang, X. Wang, W. Wei, R. Damaševičius, R. Scherer, and M. Woźniak, “Prediction of streamflow based on dynamic sliding window lstm,” *Water*, vol. 12, no. 11, 2020. [Online]. Available: <https://www.mdpi.com/2073-4441/12/11/3032>
  - [38] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2017. [Online]. Available: <https://arxiv.org/abs/1412.6980>
  - [39] L. Jing and G. Wei, “A summary of traffic flow forecasting methods,” *Journal of Highway and Transportation Research and Development*, 2004.
  - [40] G. Dang and D. Wang, “Deep recurrent stochastic configuration networks for modelling nonlinear dynamic systems,” 2024. [Online]. Available: <https://arxiv.org/abs/2410.20904>