



OPEN A combined model for short-term traffic flow prediction based on variational modal decomposition and deep learning

Chuanxiang Ren¹, Fangfang Fu¹, Changchang Yin²✉, Li Lu¹ & Lin Cheng¹

The emergence of Deep Learning provides an opportunity for traffic flow prediction. However, uncertainty and volatility exhibited by nonlinearity and instability of traffic flow pose challenges to Deep Learning models. Therefore, a combined prediction model, VMD-GAT-MGTCN, based on variational modal decomposition (VMD), graph attention network (GAT), and multi-gated attention time convolutional network (MGTCN) is proposed to enhance short-term traffic flow prediction accuracy. In the VMD-GAT-MGTCN, VMD decomposes traffic flow data to obtain the modal components, the GAT and MGTCN are integrated to design the spatio-temporal feature model to obtain the temporal and spatial features of traffic flow. The predicted value of traffic flow modal components by spatio-temporal feature model are stacked to obtain the ultimate traffic flow prediction results. The simulation experiments with the compared models and the baseline models show that the VMD-GAT-MGTCN have superior prediction accuracy and effect. It also verifies the enhancement effect of the VMD algorithm on the prediction performance of the VMD-GAT-MGTCN and the good prediction results obtained by the VMD-GAT-MGTCN in the traffic flow mutation region.

Keywords Variational modal decomposition, Graph attention network, Time convolutional networks, Short-term traffic flow prediction

In order to alleviate traffic congestion, intelligent transportation system (ITS) is widely used^{1–3}. Short-term traffic flow prediction is an important part of ITS. By accurately predicting traffic flow, the congested road sections can be identified in advance, which helps the traffic managers take timely measures to regulate traffic⁴. Furthermore, based on the predicted traffic flow, the optimization of signal timing and phase control can alleviate traffic congestion and maximize road traffic efficiency⁵. In addition, traffic flow prediction can also help travelers plan travel routes in advance, guide vehicles around congested areas, and reduce travel pressure⁶.

In order to accurately predict traffic flow, scholars have proposed many methods, which include statistical-based prediction methods^{7–9}, machine learning-based prediction methods^{10–12}, and deep learning-based prediction methods. Among these methods, deep learning-based methods can learn the laws and features hidden inside the sample data and are extensively employed. For example, Yang et al.¹³ propose an improved long short-term memory network (LSTM) to predict traffic flow through multi-scale feature enhancement and attention mechanisms. Zhang et al.¹⁴ proposed a traffic flow prediction model using prediction error as fitness value and genetic algorithm to optimize the hyperparameters of time convolutional networks (TCN).

With the deepening of the research, scholars found that only considering the time characteristics is not enough for improving the traffic flow prediction accuracy, so spatial characteristics were gradually introduced to improve the accuracy and reliability of the prediction. For example, Wang et al.¹⁵ proposed a hierarchical traffic flow prediction protocol based on spatial-temporal graph convolutional network to achieve more accurate traffic flow prediction, which integrates the spatial and temporal dependencies of intersections, and effectively predicts the traffic flow of intersections without historical data. Gao et al.¹⁶ proposed a spatio-temporal traffic flow prediction model based on graph attention network (GAT) and bidirectional gated recurrent unit (BiGRU) to obtain the spatial characteristics of road network and traffic flow data, respectively.

Meanwhile, traffic flow data has the characteristics of nonlinearity and instability, which is manifested as volatility. This presents a challenge to deep learning-based prediction methods and significantly influences the

¹College of Transportation, Shandong University of Science and Technology, Qingdao 266590, China. ²College of Electrical Engineering and Automation, Shandong University of Science and Technology, Qingdao 266590, China. ✉email: ycc2009@sdust.edu.cn

prediction results¹⁷. The signal decomposition algorithm can decompose the traffic flow time series data into several more stable intrinsic mode functions (IMFs) to diminish data volatility. This enables the prediction model to more accurately capture the temporal features of the traffic flow data. Therefore, signal decomposition algorithms are gradually applied to traffic flow prediction^{18–21}.

There are a variety of signal decomposition algorithms, such as EMD²², EEMD²³, CEEMD²⁴, and VMD²⁵. Compared with the other three algorithms, VMD can overcome the problems of modal aliasing and end-point effect, and has strong adaptive and robustness²⁶. However, the spatio-temporal features of the traffic flow exhibit a certain degree of complexity, there is still margin for research on fusing signal decomposition algorithms and deep learning methods for traffic flow prediction. Therefore, considering the spatio-temporal features and the instability of traffic flow, this paper proposes a traffic flow prediction model based on VMD and deep learning methods. The main contributions of this study include the following three aspects:

(1) A model for acquiring spatio-temporal features of traffic flow, GAT-MGTCN, is proposed, which consists of the spatial feature acquisition module (GAT) and the time feature acquisition module (MGTCN). Among them, MGTCN is designed through gated linear unit (GLU), gated tanh unit (GTU), and TCN.

(2) Based on GAT-MGTCN and VMD, a combined traffic flow prediction model is proposed, i.e., VMD-GAT-MGTCN. The model firstly decomposes the traffic flow data by VMD to obtain the traffic flow IMFs, and then IMFs are input GAT-MGTCN for traffic flow prediction, and the prediction results are obtained after stacks.

(3) Compared with the compared models based on different signal decomposition algorithms and baseline models found that the VMD-GAT-MGTCN effectively reduces the nonlinearity and instability of traffic flow data and has better prediction accuracy, especially in the traffic flow mutation region also obtains better prediction effect.

The subsequent sections of the paper are organized as follows: Sect. 2 provides an overview of the research status on short-term traffic flow prediction. Section 3 introduces the traffic flow decomposition method and the traffic flow spatio-temporal features model and establishes the VMD-GAT-MGTCN short-term traffic flow combination prediction model. Section 4 designs the experiments to validate the VMD-GAT-MGTCN and analyzes the results. Finally, in Sect. 5, this paper is summarized, and future work is suggested.

Literature review

Regarding the prediction of short-term traffic flow, statistical-based methods are firstly used. Emami et al.²⁷ proposed a Kalman filtering algorithm based on vehicle networking data for short-term traffic flow prediction on urban arterial roads. Yi et al.²⁸ developed a Kalman filter prediction model with multiple linear regressions to predict the traffic congestion time. Moshe et al.²⁹ studied 20, 40, and 60- second occupancy and volume collected at two different locations during peak hours through Box-Jenkins time series analysis and evaluated several autoregressive integrated moving average (ARIMA) models. Mai T et al.³⁰ presented an additive seasonal vector ARIMA model to predict the short-term traffic flow. And the efficiency of the proposed prediction model was evaluated by real-time traffic flow observations. Shahriari et al.³¹ proposed E-ARIMA model combining bootstrap with ARIMA model and used ARIMA and LSTM as a compared model to verify the E-ARIMA model. Chen et al.³² explored spatiotemporal mobilities of highway traffic flows and presented a multiple slope-based linear regression method to predict road travel time. Although statistical-based prediction methods offer advantages such as simple structure and rapid training, it requires high-quality training data and exhibit relatively low prediction accuracy. With the current traffic environment becoming more and more complex, statistical-based prediction methods have been difficult to meet the needs.

With the advent of machine learning, machine learning-based prediction methods have been used in traffic flow prediction. Xu et al.³³ presented a kernel K-nearest neighbor (KNN) algorithm to predict road traffic states, which maps multi-dimensional and multi-granularity road traffic state data series to high dimensions using a constructed kernel function. Wang et al.³⁴ propose a KNN prediction algorithm with asymmetric losses and overcome the limitations of traditional KNN by reconstructing Euclidean distance. Wang et al.³⁵ developed a regression framework with automatic parameter tuning for short-term traffic flow prediction, which used support vector machine (SVR) and Bayesian optimization for regression model and parameter selection, respectively. Hu et al.³⁶ used SVR to build the traffic flow prediction model, in which model parameters are optimized by particle swarm algorithm. Su et al.³⁷ introduced an incremental SVR-based short-term traffic flow prediction model that updates the prediction function in real-time through incremental learning techniques. Zhu et al.³⁸ proposed a linear conditional Gaussian Bayesian network model to predict traffic flow, which considers both spatio-temporal and speed information in the traffic flow. Sun et al.³⁹ Wang et al. proposed a fully automatic dynamic process KNN method for traffic flow prediction, which can automatically adjust KNN parameters with good robustness. Ryu et al.⁴⁰ developed a short-term traffic flow prediction method that uses mutual information to construct traffic state vectors and processed the traffic state vectors using the KNN model. Lin et al.⁴¹ combined the SVR and KNN for short-term traffic flow prediction. The above-cited machine learning-based traffic flow prediction methods can obtain excellent prediction accuracy. However, the inability to characterize complex nonlinearity spatio-temporal correlations of traffic flow largely restricts the applicability of such prediction models.

Deep learning methods with the advantages of strong robustness and good fault tolerance have been used more and more frequently in the traffic flow prediction. Kipf et al.⁴² proposed graph convolutional network (GCN) for traffic flow prediction, which solves the shortcomings of convolutional neural networks (CNN) that can't deal with topological maps. Koesdwiady et al.⁴³ took weather as influencing factor, and designed the deep belief networks model to predict traffic flow. Sun et al.⁴⁴ proposed a short-term traffic flow prediction model based on an improved GRU and added a bidirectional feedback mechanism to extract spatio-temporal features to further improve prediction accuracy. Tian et al.⁴⁵ presented an approach based on LSTM to predict traffic flow with missing data, which used multi-layered temporal smoothing to deduce missing data. Yang et al.¹³ enhanced the LSTM model by introducing the attention mechanism to capture key traffic flow values, which are

closely associated with the current time step. Li et al.⁴⁶ developed a GCN model, which improves the accuracy of medium and long-term traffic prediction through adaptive mechanism and multi-sensor data correlation convolution block.

With the growing complexity of traffic flow prediction and the increasing number of factors to be considered, single deep learning prediction models mentioned above struggle to meet prediction requirements, and the combined deep learning prediction models have become a major research direction of traffic flow prediction. Chen et al.⁴⁷ proposed a method based on dynamic space-time graph-based CNN to predict the traffic flow. In the method, the time characteristics are extracted by the spatiotemporal convolution layer based on graph, and the dynamic graph structure is predicted by the graph prediction flow. Qiao et al.⁴⁸ proposed a combination prediction model for the short-term traffic flow using the LSTM and 1DCNN and made use of the resulting spatio-temporal characteristics for regression prediction. Sun et al.⁴⁹ proposed a selective stacking GRU model for road network traffic volume prediction, which uses linear regression coefficient for spatial pattern mining and stacked gated cycle units for multi-road traffic prediction. Peng et al.⁵⁰ presented a long-term traffic flow prediction method, in which dynamic traffic flow probability graph is used to model traffic network, GCN is used to learn spatial features, and LSTM unit is used to learn temporal features. Hu et al.⁵¹ propose an intelligent dynamic spatiotemporal graph convolution system for traffic flow prediction to improve prediction performance by capturing complex and dynamic spatial and temporal dependencies. Yu et al.⁵² proposed a traffic flow prediction method, which employed principal component analysis to analyze intersection correlation, and used convolutionally GRU and bidirectional gated recurrent units (BiGRU) to extract spatio-temporal characteristics of traffic flow. Liu et al.⁵³ proposed a self-attention-based model combining CNN, LSTM, and attention mechanism to predict the short-term traffic flow. Wang et al.⁵⁴ considered the impact of weather on the traffic flow and proposed a combined prediction model which incorporates attention mechanism, 1DCNN, and LSTM. Zhang et al.⁵⁵ presented a combined traffic flow prediction model, which used an improved graph convolution recurrent network based on residual connected blocks and LSTM to automatically extract spatio-temporal features of traffic flow.

Although deep learning-based prediction methods have further improved the prediction accuracy, there is still the problem of not being able to comprehensively extract the features contained of the traffic flow data. The signal decomposition algorithm provides a way to solve this problem, which can decompose the original signal (data) into multiple IMFs, making it easier to fit the complex nonlinear relationships, thus improving the accuracy of data prediction. Among the various signal decomposition algorithms, VMD is used to traffic flow prediction because of its ability to overcome the problems of mode aliasing and end-point effect. Liu et al.⁵⁶ proposed a combined prediction model for traffic flow prediction, which includes VMD, the group method of data handling neural network, BiLSTM network, and ELMAN network. And the imperialist competitive algorithm is used to optimize the combined prediction model. Zhao et al.⁵⁷ proposed a combined traffic flow prediction model based on VMD and improved dung beetle optimization-long short term memory network. Jing et al.⁵⁸ introduced a model based on short-term memory network to predict traffic flow, which employs VMD to tackle modal aliasing and enhance the prediction precision. The above-mentioned traffic flow prediction methods utilizing VMD achieved better prediction results. However, these methods primarily focus on the temporal features of traffic flow and ignore the influence of spatial features on the accuracy of prediction. In this paper, the spatio-temporal features of the traffic flow are considered, and the VMD, as well as GCN and TCN are used to design a combined model for the short-term traffic flow prediction.

Methodology

Traffic flow prediction model

The real traffic flow is nonlinear, unstable, and highly dependent on time and space, which requires the prediction model to overcome this difficulty and fully explore its time and space characteristics. VMD can decompose the traffic flow data and make its time features easy to obtain as well as reduce the instability in the data.

TCN is a novel architecture based on CNN, which not only has the characteristics of parallelism and causality, but also has a flexible receptive field, and is suitable for processing time series data such as traffic flow. Based on TCN and the gating mechanism, the traffic flow time feature acquisition module is designed, namely multi-gated time convolution network (MGTCN). GAT is obtained by introducing the attention mechanism into GCN. GCN can capture complex patterns and nonlinear relationships in the graph structure data. The attention mechanism enables the GAT to capture the complex spatial relationships among nodes, accurately express the dependency and the degree of influence between nodes by learning the attention weights and extract the nonlinear relationships between nodes. And GAT is used for the acquisition of traffic flow spatial features.

Based on the MGTCN and GAT, the traffic flow spatio-temporal feature model is designed and named GAT-MGTCN. On the basis of the VMD and the GAT-MGTCN, a combined traffic flow prediction model is proposed and named VMD-GAT-MGTCN. In the model, VMD is used to decompose traffic flow data and obtain the IMFs in the data, while the GAT-MGTCN extracts temporal and spatial features. From the perspective of layers, VMD-GAT-MGTCN consists of input layer, hidden layer, and output layer. The overall framework is shown in Fig. 1.

The input layer provides input data. The obtained raw traffic flow data includes the time series data of traffic flow and the spatial data of each node. After preprocessing, including data cleaning, normalization, etc., the time series data of traffic flow is obtained, and then is decomposed by VMD to obtain IMFs. Finally, the IMFs are combined with spatial data to form the input data.

The hidden layer includes the spatio-temporal feature layer and the Dropout layer. The spatio-temporal feature layer is composed of the GAT-MGTCN. GAT receives IMFs and spatial data from the input layer, constructs an adjacency matrix of the fully connected graph to represent the dependency between nodes, and forms a feature matrix with IMFs. The sliding window method is used to convert the data into a time series

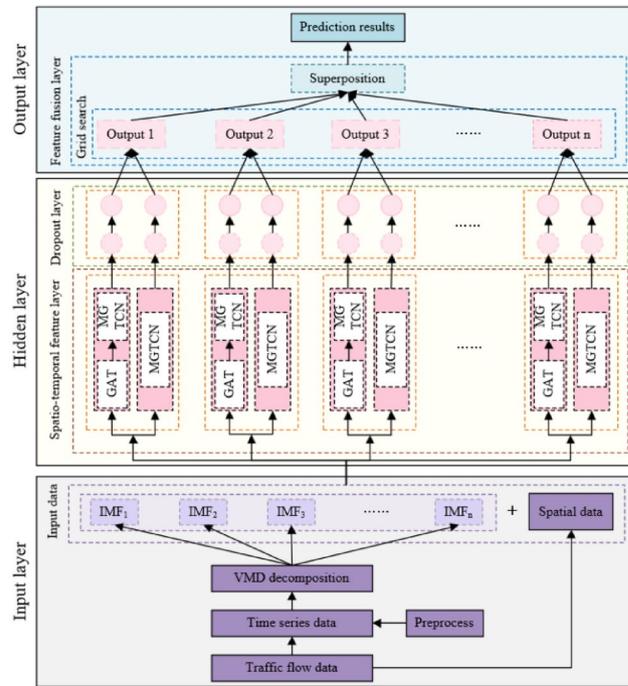


Fig. 1. Overall framework of the VMD-GAT-MGTCN.

samples, and the feature dimension of each sample is the sum of the number of modal components and spatial features. The time series samples were input into MGTCN for time feature extraction, and a set of prediction data was obtained. At the same time, only IMFs were input into MGTCN to extract the time feature, and another set of prediction data was obtained. The two sets of prediction data are processed through the Dropout layer to prevent overfitting of the model.

Output layer includes feature fusion layer and prediction results. Feature fusion layer assigns appropriate weights to the outputs of the Dropout layer through a grid search algorithm and performs superposition. Finally, the superimposed results as predicted results are output.

Traffic flow decomposition method

VMD possesses the characteristics of adaptive and completely non-recursive, which allows for setting the number of IMFs based on the actual situation⁵⁹. Compared with EMD, VMD employs a variational decomposition method that enables adaptive segmentation of each component within the frequency domain of the signal. The phenomenon of mode aliasing generated by the recursive decomposition of EMD is resolved, enhancing noise robustness and reducing end-point effect. Therefore, VMD is selected as the signal decomposition algorithm to decompose traffic flow data. The essence of VMD is to construct and solve variational decomposition problem, and the specific process is as follows.

(1) Construct variational problem. Assuming that the original signal $S(t)$ needs to be decomposed into M IMFs with finite bandwidth and center frequency, the signal decomposition problem can be turned into a constrained variational problem, i.e., the sum of the estimated bandwidths of IMFs is minimized and the sum of IMFs is equal to the original signal, and the corresponding formulars are as follows:

$$\begin{cases} \min_{\{u_m\}, \{\omega_m\}} \left\{ \sum_m \left\| \partial_t \left[\left(\delta(t) + \frac{m}{\pi t} \right) * u_m(t) \right] e^{-m\omega_m t} \right\|_2^2 \right\} \\ s.t. \sum_{m=1}^M u_m = S(t) \end{cases} \quad (1)$$

where $\{u_m\}$ represent the m th modal component, $\{\omega_m\}$ represent the m th center frequency after decomposition, $\delta(t)$ is the Dirac function, and $*$ denotes the convolution operation.

(2) Solve the variational problem. Lagrange multipliers and quadratic penalty factors are introduced in Eq. (2) to solve the optimal solution for the constrained variational problem. Equation (2) is transformed into an unconstrained variational problem, and the corresponding formular is as follows:

$$L(\{u_m\}, \{\omega_m\}, \lambda) = \alpha \sum_m \left\| \partial_t \left[\left(\delta(t) + \frac{m}{\pi t} \right) * u_m(t) \right] e^{-m\omega_m t} \right\|_2^2 + \left\| S(t) - \sum_k u_m(t) \right\|_2^2 + \left\langle \lambda(t), S(t) - \sum_m u_m(t) \right\rangle \quad (2)$$

Input: The original signal: $S(t)$
The number of modes: M
The quadratic penalty factors: α
The Lagrange multiplier update rate: τ
The convergence tolerance: tol
The maximum number of iterations: N

Output: The m th modal component: $\{u_m\}$
The m th center frequency: $\{\omega_m\}$

- 1: **Initialize** $\{\hat{u}_m^1\}$, $\{\hat{\omega}_m^1\}$, and $\hat{\lambda}^1$
- 2: **for** $\square = 1$ to \square **do**
- 3: **for** $m = 1$ to M **do**
- 4: update \hat{u}_m for all $\omega \geq 0$:
- 5:
$$\hat{u}_m^{n+1}(\omega) \leftarrow \frac{\hat{f}(\omega) - \sum_{i=1, i \neq m}^M \hat{u}_i^{n+1}(\omega) - \sum_{i=1, i > m}^M \hat{u}_i^n(\omega) + \frac{\hat{\lambda}^n(\omega)}{2}}{1 + 2\alpha(\omega - \hat{\omega}_m^n)^2}$$
- 6: update $\hat{\omega}_m$:
- 7:
$$\hat{\omega}_m \leftarrow \frac{\int_0^\infty |\hat{u}_m(\omega)|^2 \cdot \omega d\omega}{\int_0^\infty |\hat{u}_m(\omega)|^2 d\omega}$$
- 8: **end for**
- 9: update $\hat{\lambda}^{n+1}$ for all $\omega \geq 0$:
- 10:
$$\hat{\lambda}^{n+1}(\omega) \leftarrow \hat{\lambda}^n(\omega) + \tau[\hat{f}(\omega) - \sum_{m=1}^M \hat{u}_m^{n+1}(\omega)]$$
- 11: check for convergence:
- 12: **if** $\sum_{m=1}^M \left\| \hat{u}_m^{n+1}(\omega) - \hat{u}_m^n(\omega) \right\|_2 / \left\| \hat{u}_m^n(\omega) \right\|_2 < \varepsilon$ **then**
- 13: **break**
- 14: **end if**
- 15: **end for**
- 16: **return** $\{u_m\}$, $\{\omega_m\}$

Algorithm 1. VMD.

where the $\lambda(t)$ is Lagrange multipliers, α is quadratic penalty factors and its value represents the degree of suppression of Gaussian noise interference.

The pseudo-code of VMD is presented in Algorithm 1.

Traffic flow Spatio-Temporal feature model

Aiming to extract the spatio-temporal feature inherent in traffic flow data, a spatio-temporal feature model, GAT-MGTCN, is designed, which consists of spatial feature acquisition module and time feature acquisition module.

The structure of traffic flow Spatio-Temporal feature model

Based on the GAT and MGTCN, the GAT-MGTCN is obtained, which includes input layer, hidden layer, fusion layer, and output layer, and its structure is shown in Fig. 2.

The input layer comprises the IMFs of traffic flow data and its associated spatial data.

The hidden layer is used to extract spatio-temporal feature in traffic flow data, which consists of the spatial feature layer and temporal feature layer, namely GAT and MGTCN. Furthermore, GAT and MGTCN are combined to extract temporal and spatial features, MGTCN alone is used to extract temporal features, and then their results are fused in the feature fusion layer. This can avoid the insufficient time feature extraction and ensures deep extraction of temporal and spatial dependencies by treating spatial and temporal features separately, which together improve the overall performance of the model.

In the feature fusion layer, the grid search method is used for assigning weights to upper-layer outputs to facilitate stacking.

Finally, the output layer presents the calculation results of the spatio-temporal feature model.

Spatial feature acquisition module

The spatial feature acquisition module, GAT, is obtained by introducing the attention mechanism into GCN, which can dynamically learn the importance weights of neighbor nodes.

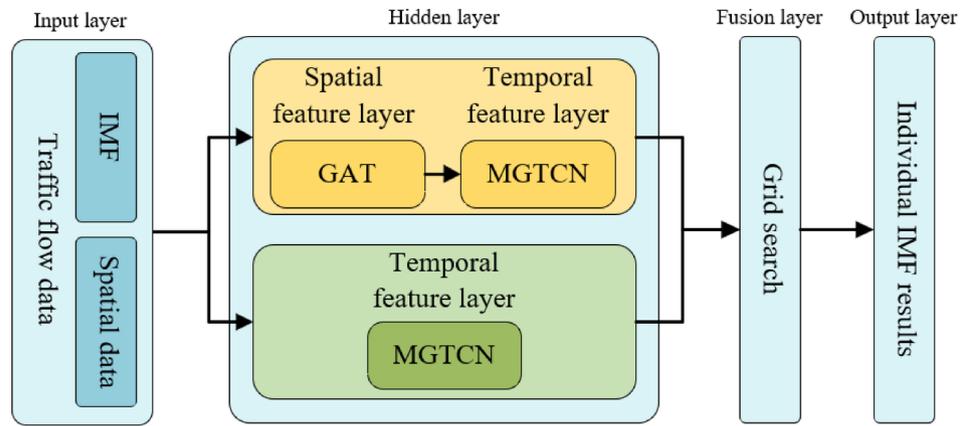


Fig. 2. Structure diagram of the spatio-temporal feature model.

1) *GCN* GCN regards the graph structure as an undirected graph, firstly obtain the node information near the target node, and then weight calculation on the information to update and learn the features of the target node, so as to complete the following prediction task⁶⁰. Furthermore, the introduction of nonlinear transformations through the activation function *ReLU* can help the GCN capture complex patterns and nonlinear relationships in the data.

The formulas for the GCN are given in Eq. (3) and Eq. (4).

$$H^1 = \sigma \left(D^{-1/2} (A + I) D^{-1/2} X \omega^1 \right) \quad (3)$$

$$H^i = \sigma \left(D^{-1/2} (A + I) D^{-1/2} H^{i-1} \omega^i \right) \quad (4)$$

where H^1 is the feature output of the first layer of the GCN, H^i is the feature output of i th layer of the GCN, σ denotes the Sigmoid activation function, A denotes the adjacency matrix of the graph, I is the unit-diagonal matrix, D is the degree matrix of $A + I$, X denotes the feature vector of the graph, ω^1 denotes the connection weights of the first layer, and ω^i denotes the connection weights of the i th layer.

Since the adjacency matrix A does not take into account its weighting, $A + I$ is used as the adjacency matrix of the graph in the GCN. And in order to alleviate the problems of numerical dispersion and gradient explosion that occur during the learning process, the adjacency matrix is subjected to a normalization operation using $D^{-1/2} (A + I) D^{-1/2}$.

2) *Attention mechanism* The emergence of the attention mechanism alleviates the problem of neural networks failing to obtain optimal results due to insufficient arithmetic power. The core idea of the attention mechanism is to assign a weight value to each element in the input feature. Through this mechanism, the model can pay more attention to features with significant impact on the prediction results, while reduce the attention to that having a smaller impact, and exclude irrelevant features to enhance the ability of the model to extract features⁶¹.

3) *GAT* GAT is derived from GCN and the attention mechanism. Especially, in order to steady the learning process and refine learning accuracy, GAT adopted a multi-head attention mechanism, that is, m independent attention mechanisms are used to calculate the hidden state of nodes, concatenate their feature, and finally obtain a new feature vector of nodes by calculating the average value of the feature. This enables the GAT to capture the complex spatial relationships among nodes, accurately express the dependency and the degree of influence between nodes by learning the attention weights, and extract the nonlinear relationships between nodes, so as to enhance GAT the acquisition of spatial features.

In GAT, the graph attention layer is the core component and its structure diagram is shown in Fig. 3. The calculation process of the graph attention layer includes the training of weight matrix, the calculation of the attention coefficient between nodes, and the update of the feature vector.

①The training of weight matrix. The weight matrix represents the relationship between the sets of input and output node feature vectors, and the corresponding formula is as follows:

$$\vec{h} = \omega \vec{h} \quad (5)$$

Where ω is the weight matrix, \vec{h} is the set of input node feature vectors, and \vec{h} is the set of output node feature vectors.

②The calculation of the attention coefficient between nodes. The attention coefficient between nodes is calculated as follows:

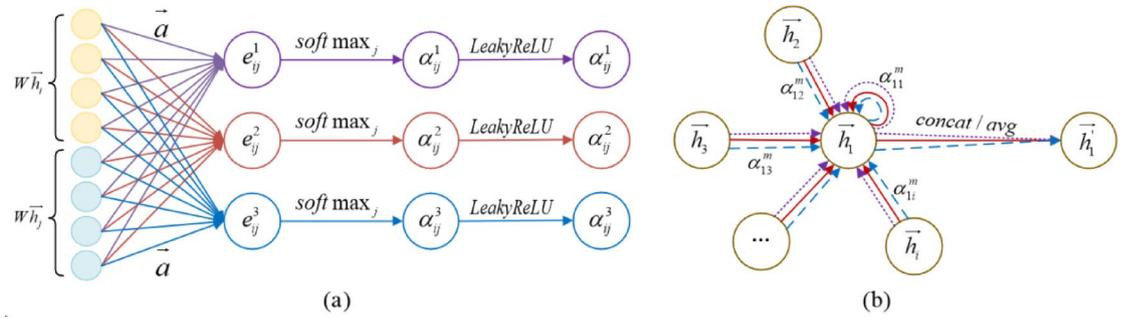


Fig. 3. Structure diagram of graph attention layer. **(a)** Calculation of attention coefficient of multi-head, **(b)** Schematic diagram of multi-head attention mechanism with 3 heads.

Input: The graph: $G(V, E)$
 The set of node features: $h = \{h_i | i \in V\}$
 The number of attention heads: M
 The weight matrix for each head: ω^m
 The neighboring nodes of node: N_i

Output: The set of output node feature vectors: $h' = \{h'_i | i \in V\}$
 The attention coefficients: α_{ij}

- 1: **Initialize** For each node $i \in V$, set initial node features h_i
- 2: **for** each attention head $m = 1$ to M **do**
- 3: **for** each node $i \in V$ **do**
- 4: compute transformed node features:
- 5: $h_i^m \leftarrow \omega^m h_i$
- 6: **for** each neighbor node $j \in N_i$ **do**
- 7: compute the importance of nodes j to i :
- 8: $e_{ij}^m \leftarrow \text{LeakyReLU}(a^T [\omega^m h_i \parallel \omega^m h_j])$
- 9: **end for**
- 10: normalize attention coefficient:
- 11: $\alpha_{ij}^m \leftarrow \frac{\exp(e_{ij}^m)}{\sum_{k \in N_i} \exp(e_{ik}^m)}$
- 12: aggregate neighbor features:
- 13: $h_i^m \leftarrow \sum_{j \in N_i} \alpha_{ij}^m \omega^m h_j^m$
- 14: **end for**
- 15: **end for**
- 16: average the outputs from all attention heads:
- 17: $h'_i \leftarrow \sigma(\frac{1}{M} \sum_{m=1}^M \sum_{j \in N_i} \alpha_{ij}^m \omega^m h_j)$
- 18: **return** $h' = \{h'_i, i \in V\}, \alpha_{ij}$

Algorithm 2. GAT.

$$\alpha_{ij} = \text{softmax}(x_j e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k \in N_i} \exp(e_{ik})} \tag{6}$$

where α_{ij} denotes the attention coefficient from nodes i to j , softmax is a normalized function, e_{ij} denotes the importance of nodes j to i , and \exp is the exponential function.

In Eq. (7), the calculation formula of e_{ij} is as follows:

$$e_{ij} = \text{att}(\omega \vec{h}_i, \omega \vec{h}_j) \tag{7}$$

Add LeakyReLU nonlinear function to Eq. (8) for activation, and the attention coefficient formula is fully expanded and can be expressed as:

$$\alpha_{ij} = \frac{\exp(\text{Leaky Re lu}(a^T [\omega \vec{h}_i \parallel \omega \vec{h}_j]))}{\sum_{k \in N_i} \exp(\text{Leaky Re lu}(a^T [\omega \vec{h}_i \parallel \omega \vec{h}_k]))} \tag{8}$$

where a^T denotes the transpose matrix of the forward propagation weight vector matrix of the single-layer neural network, N_i denotes the neighboring nodes of node i , and \parallel denotes the function that concatenation node i and j .

③The update of feature vector. The weighted summation and average value of the features are calculated to obtain the new feature vector of node i , and the formular is as follows:

$$\vec{h}_i = \sigma \left(\frac{1}{M} \sum_{m=1}^M \sum_{j \in N_i} \alpha_{ij}^m \omega^m \vec{h}_j \right) \tag{9}$$

where σ is the Sigmoid function, α_{ij}^m is the normalized attention coefficient of the m th head attention mechanism, and ω^m is the weight matrix of the corresponding linear transformation.

The pseudo-code of GAT is presented in Algorithm 2.

Time feature acquisition module

Based on TCN and the gating mechanism, the traffic flow time feature acquisition module is designed, namely MGTCN.

1) TCN: TCN is a novel architecture based on CNN and consists of dilated causal convolution network and residual blocks⁶². The structure diagram of TCN is shown in Fig. 4. Specifically, Fig. 4 (a) demonstrates the dilated causal convolution network with two hidden layers, dilation factors d is 1, 2, 4, and filter size is 2. In Fig. 4(b), a 1×1 convolution is introduced to the residual path to ensure a consistent number of features. Dilated causal convolution networks with fewer layers provide TCN with a larger receptive field. The residual block can learn the constant mapping function, so that the neural network can transfer information across the layers and avoid gradient vanishing and explosion. Furthermore, TCN, which not only has the characteristics of parallelism and causality, but also has a flexible receptive field, is suitable for processing time series data such as traffic flow.

In dilated causal convolution network, for a one-dimensional sequence input $X \in R^n$ and a filter $f : \{0, \dots, k - 1\}$, the dilated convolution operation is calculated as follows:

$$F(s) = X * f = \sum_{l=0}^{k-1} f(l)X_{s-d \times l} \tag{10}$$

Where s is element of the sequence, $F(s)$ denotes the result of dilated convolution operation, X denotes the input data, f denotes the filter for the convolution operation, d denotes the dilation rate, and l is the number of layers in which the convolution kernel is located.

Residual blocks include dilated causal convolution network, weight normalization, $ReLU$ activation functions, and Dropout. Dilated causal convolution network expands the receptive field by introducing dilation into the convolution kernel while preserves causality, which helps to capture long-term dependencies in time series. Weight normalization improves training stability and generalization ability. $ReLU$ introduces nonlinear

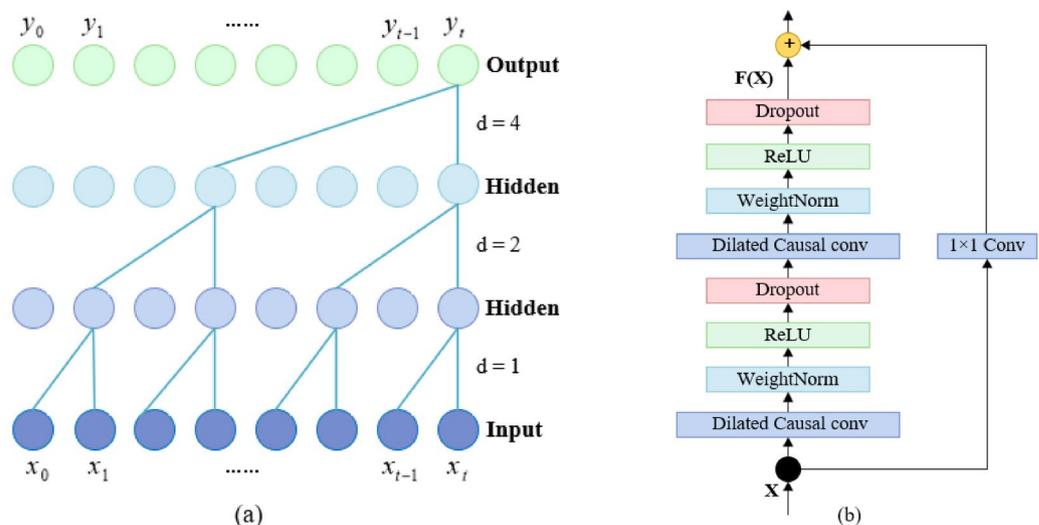


Fig. 4. Structure diagram of TCN. (a) Dilated causal convolution network, (b) TCN residual block.

characteristics to enable the network to learn complex nonlinear relationships. Dropout helps prevent overfitting and makes the network more robust.

The residual block adds the original input X of the model to the constant mapping function $F(X)$ to obtain the final output P , which is calculated as follows:

$$P = \text{ReLU} (X + F(X)) \tag{11}$$

2) *MGTCN* Adding gating mechanisms to neural networks can control the flow of information and features, thereby improving the convergence speed and computational efficiency of the networks⁶³. Based on this, *MGTCN* is designed, which consists of two TCNs with the attention mechanism (TCN-ATT), two GLU gating mechanisms, and one GTU gating mechanism. And the structure diagram of *MGTCN* is shown in Fig. 5. The *MGTCN* can extract and fuse temporal features at different time scales through multiple gating mechanisms, which improves the capability to capture temporal features.

In *MGTCN*, attention mechanism enhances the ability of TCN to acquire temporal features of traffic flow. The processing of double GLU gating mechanisms for the results of two TCNs with attention mechanism can increase nonlinear features and filter useless messages. Then the two processed results are stacked by assigning weights through the attention mechanism. Finally, the stacked results and the original data are processed through the GTU gating mechanism to determine the features to be retained or discarded, and the final output results are obtained. The formula for the *MGTCN* is as follows:

$$h = \tanh (\omega_1 * X_1) \cdot \sigma (\omega_2 * X_2) \tag{12}$$

where h is the output, \cdot denotes the multiplication operation of the matrix, $*$ denotes the convolution operation, ω_1 and ω_2 are the parameters that need to be learned, σ is *Sigmoid* function used to filter useless information, X_1 denotes the input data, and X_2 is the output obtained by double GLU gating.

And X_2 can be calculated as follows:

$$X_2 = \omega_7 ((\omega_3 * X_3) \cdot \sigma (\omega_4 * X_4)) \cdot \omega_8 (\sigma (\omega_5 * X_3) \cdot (\omega_6 * X_4)) \tag{13}$$

where $\omega_3, \omega_4, \omega_5, \omega_6, \omega_7,$ and ω_8 are the model parameters to be learned, X_3 denotes the output result of TCN-Att1, and X_4 is the output result of TCN-Att2.

The pseudo-code of *MGTCN* is presented in Algorithm 3.

Experiments and analysis of results

Experimental data

The publicly available dataset PeMS04 is selected for the experiments to validate the VMD-GAT- *MGTCN* model. The dataset contains data collected by 307 sensors on the freeway system in the Los Angeles, California area at five-minute intervals. In PeMS04, the traffic flow data of 14 consecutive days from January 1 to January 14, 2018 are selected as the experimental dataset after processing for redundant, abnormal, and missing data, where the first 12 days' data constitute the training set, while the remaining 2 days' data form the test set. In addition, to ensure that the selection of data points is unbiased, the traffic flow prediction for data collection point 109 was randomly chosen as an example for experimental analysis, serving to evaluate the model's prediction performance.

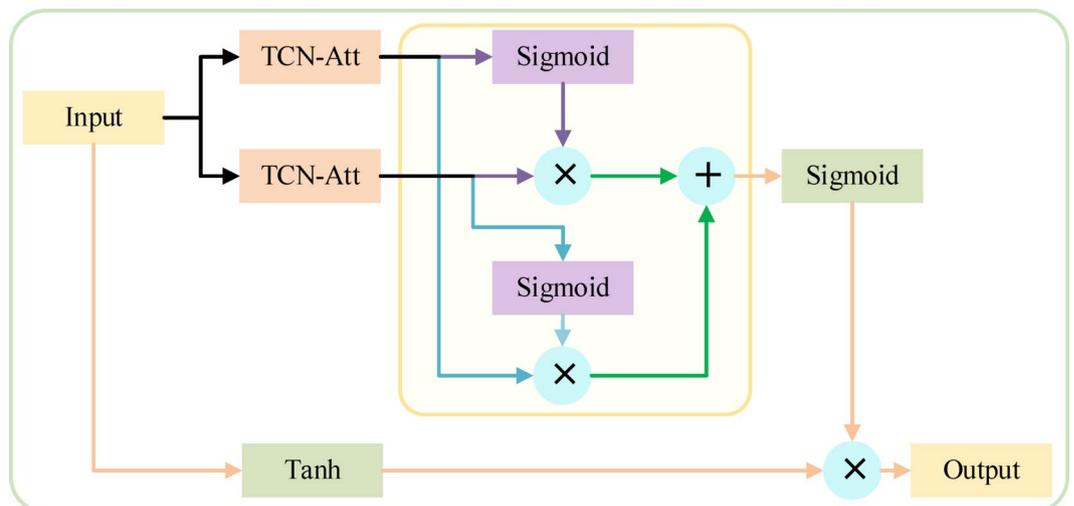


Fig. 5. Structure diagram of *MGTCN*.

Input: The input data: X
The filter: $f: \{0, \dots, k-1\}$
The dilation rate: d
The number of layers: l

Output: The final output: P

- 1: **Initialize:**
- 2: set initial input: $X \leftarrow X_0$
- 3: **for** each layer $l = 1$ to l **do**
- 4: **for** each element s in sequence X **do**
- 5: compute dilated convolution output:
- 6: $F(s) = \sum_{l=0}^{k-1} f(l)X_{s-d \times l}$
- 7: **end for**
- 8: apply non-linearity:
- 9: $X_1 \leftarrow \text{ReLU}(X + F(X))$
- 10: **end for**
- 11: compute attention-based TCN outputs:
- 12: $X_3 \leftarrow \text{TCN} - \text{ATT1}$
- 13: $X_4 \leftarrow \text{TCN} - \text{ATT2}$
- 14: combine attention outputs with gating:
- 15: $X_2 \leftarrow \omega_7((\omega_3 * X_3) \cdot \sigma(\omega_4 * X_4)) \cdot \omega_8(\sigma(\omega_5 * X_3) \cdot (\omega_6 * X_4))$
- 16: apply GTU gating on combined outputs:
- 17: $P \leftarrow \tanh(\omega_1 * X_1) \cdot \sigma(\omega_2 * X_2)$
- 18: **return** final output: P

Algorithm 3. MGTCN.

Software and hardware configuration	Configuration parameter
CPU	CPU Intel i5-13500 H@2.60 GHz
RAM	16G
Programming language	Python 3.8.8
Deep learning framework	PyTorch 1.10.2

Table 1. Experimental environment.

Experimental environment and model evaluation index

The hardware and software conditions of the experimental environment are provided in Table 1.

The mean absolute error (MAE), the root mean square error (RMSE), the mean absolute percentage error (MAPE), and the coefficient of determination (R^2) are selected as the evaluation indexes. MAE reflects the error between the predicted value and the actual value, RMSE reflects the deviation of the predicted value from the actual value, MAPE is the error after eliminating the influence of proportional effect, and R^2 reflects the correlation of linear strength between the predicted value and the actual value. The calculation formulas of the three evaluation indexes are as follows:

$$MAE = \frac{1}{M} \sum_{m=1}^M |Y_T - Y_D| \quad (14)$$

$$RMSE = \sqrt{\frac{1}{M} \sum_{m=1}^M (Y_T - Y_D)^2} \quad (15)$$

$$MAPE = \frac{1}{M} \sum_{m=1}^M \left| \frac{Y_T - Y_D}{Y_T} \right| \quad (16)$$

$$R^2 = 1 - \frac{\sum_{m=1}^M (Y_D - Y_T)^2}{\sum_{m=1}^M (Y_D - \bar{Y}_D)^2} \quad (17)$$

where M is the total number of samples, Y_T is the predicted value, Y_D is the actual value of the sample, and \bar{Y}_D is the mean of the actual values.

Model training and experimental results

(1) Model Parameter Settings.

1) VMD parameter settings In VMD, the penalty factor is set to 2000, the noise tolerance is 0.1, and the uniform initialization coefficient is 1. In addition, there is a crucial parameter in VMD, i.e. the number of signal decomposition. If it is too large, modal aliasing may occur, meaning that modes of different frequencies are insufficiently separated, resulting in overlap in the frequency domain. If it is too small, the complexity of the signal may not be fully captured, leading to an incomplete representation of the original signal and the loss of critical information in the reconstructed signal. To determine its optimal value, all other parameters were held constant, and the numbers of IMFs varied from a lower value of 3 and incrementally increasing to 10. Experiments were conducted at different numbers of IMFs, and the results are presented in Table 2.

As can be seen in Table 2, with the increase in the number of IMFs, the center frequency of the last IMF exhibits an upward trend and starts to stabilize from a value of 8. To avoid the phenomenon of modal aliasing, the number of signal decompositions is therefore selected as 8.

2) Deep Learning Hyperparameter Settings In the GAT model, the multi-head attention mechanism is set to 3 heads, and the number of the neural units in the hidden layer is 64. For the MGTCN model, the first layer of TCN contains 128 neural units with expansion factors of [1, 2, 4], while the second hidden layer contains 64 neural units with expansion factors of [1, 2, 4]. Adam optimization algorithm is employed with an initial learning rate of 0.001. The training process involves 100 rounds, utilizing the mean square error function as the loss function, and a training batch size of 32. These parameter settings were obtained from several experiments. In the experiments, various parameter combinations were tested, and the optimal settings were determined based on the model's performance on the test set.

(2) Results and analysis The traffic flow time series data are input into the VMD-GAT-MGTCN, and was decomposed firstly into 8 IMFs through VMD. The 8 IMFs and their corresponding center frequency were obtained, as shown in Fig. 6. It can be seen that each IMF clearly captures a specific frequency in the traffic flow time series data, and different IMFs are relatively independent from each other, without modal aliasing and phenomena. In addition, the various IMFs are smooth and have no obvious high-frequency components, and there is no end effect. This indicates that the VMD decomposition is effective and accurate.

Then, the IMFs are integrated with the spatial data of traffic flow to form the input data. The input data is processed through hidden layer and stacked by feature fusion layer in VMD-GAT-MGTCN, and the prediction results are obtained. The prediction results on the test set can be seen in Fig. 7. Further, the error analysis of the prediction results is performed, which yields the evaluation index MAE, RMSE, MAPE, and R^2 as presented in Table 3.

From Fig. 7, it can be found that results of the VMD-GAT-MGTCN fit the actual traffic flow remarkably well and also match the trend of the actual traffic flow. This indicates that the VMD-GAT-MGTCN employing VMD for decomposing the traffic flow data and considering the spatio-temporal features of traffic flow data and achieves a better prediction result. Furthermore, it can be found from Table II that the values of the three evaluation indexes are small, which illustrates the excellent predictive efficacy of the VMD-GAT-MGTCN.

Signal decomposition algorithm experiment and analysis

To verify the effect of VMD on the prediction performance of the VMD-GAT-MGTCN, the spatio-temporal feature model, GAT-MGTCN, is selected as the compared model, i.e., the traffic flow data are directly input into

IMFs	Number							
	3	4	5	6	7	8	9	10
IMF1	0.0321	0.0276	0.0285	0.0263	0.0261	0.0260	0.0261	0.0250
IMF2	0.0370	0.0343	0.0324	0.0326	0.0319	0.0297	0.0271	0.0197
IMF3	0.3453	0.0873	0.1364	0.0339	0.0299	0.0365	0.0367	0.0402
IMF4		0.1812	0.2600	0.1001	0.0973	0.0636	0.0540	0.0406
IMF5			0.3188	0.1811	0.1697	0.1652	0.1347	0.0876
IMF6				0.3564	0.2439	0.2959	0.2012	0.1301
IMF7					0.3628	0.3977	0.3100	0.1768
IMF8						0.4608	0.3998	0.2584
IMF9							0.4656	0.3582
IMF10								0.4660

Table 2. Decomposition results for different numbers of IMFs.

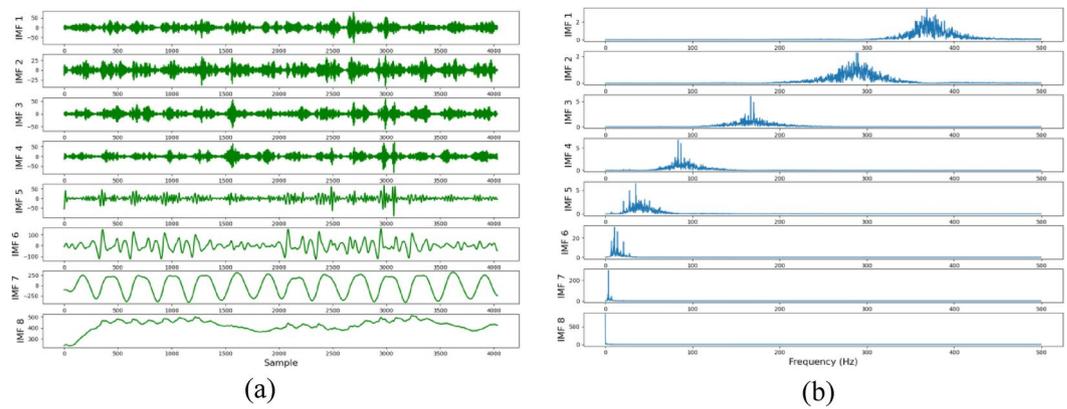


Fig. 6. VMD decomposition results. (a)IMFs of traffic flow time series data, (b) Center frequencies of each IMF.

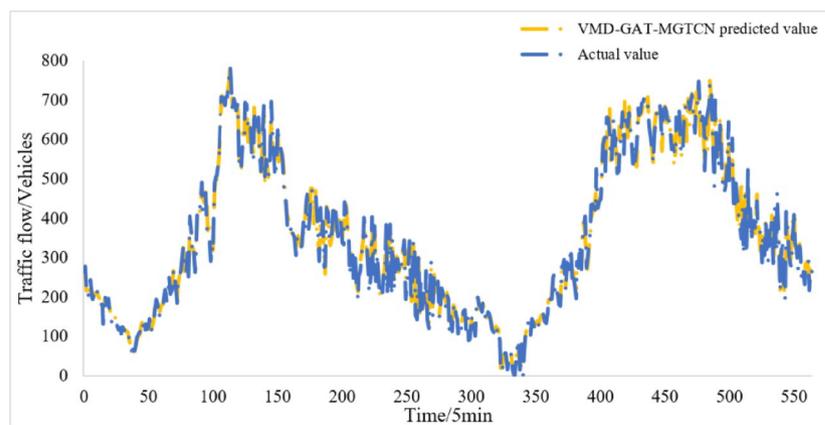


Fig. 7. Prediction results of the VMD-GAT-MGTCN.

Model	MAE	RMSE	MAPE (%)	R ²
VMD-GAT-MGTCN	15.59	21.64	11.75	0.9853

Table 3. Evaluation indexes value of the VMD-GAT-MGTCN.

GAT-MGTCN for traffic flow prediction without VMD decomposition. Furthermore, three types of commonly used signal decomposition algorithms EMD, EEMD, and CEEMD are selected, and based on the VMD-GAT-MGTCN, the compared models EMD-GAT-MGTCN, EEMD-GAT-MGTCN, and CEEMD-GAT-MGTCN are designed for traffic flow prediction experiments.

1) *Model parameter settings* The average decomposition number for signals in EMD, EEMD, and CEEMD is set to 100, and in EEMD and CEEMD, the ratio of standard deviation of additional noise to standard deviation of signal to be decomposed is 0.2. The initial learning rate, training rounds, loss function, and training batch size of the compared models are the same as the VMD-GAT-MGTCN.

2) *Results and analysis* The compared models, GAT-MGTCN, EMD-GAT-MGTCN, EEMD-GAT-MGTCN, and CEEMD-GAT-MGTCN are trained and tested under the same dataset, and the prediction results for the test set are obtained, as shown in Fig. 8. In addition, to facilitate a more effective comparison between the VMD-GAT-MGTCN and other models, the prediction results of the VMD-GAT-MGTCN are also shown in Fig. 8.

Figure 8 illustrates that both the VMD-GAT-MGTCN and the compared models maintain a high degree of overlap with the actual traffic flow values, and yield good prediction results. In order to more clearly observe the differences in the prediction results of each model, the two consecutive hours' results are randomly selected and displayed in Fig. 9. The Fig. 9 clearly indicates that compared with GAT-MGTCN, EMD-GAT-MGTCN, EEMD-

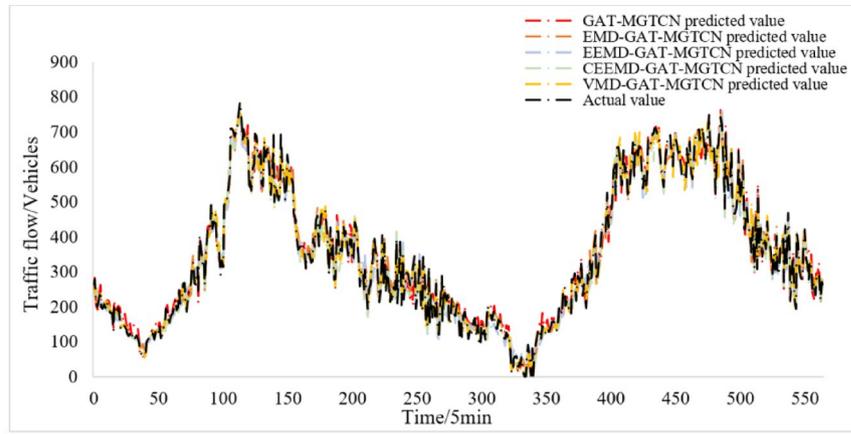


Fig. 8. Prediction results of the compared models.

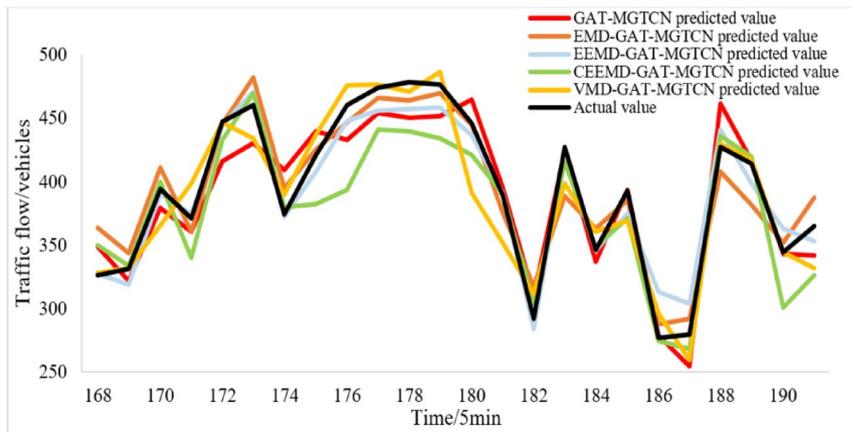


Fig. 9. Prediction results of the compared models for two hours.

Model	MAE	RMSE	MAPE (%)	R ²
GAT-MGATCN	20.87	28.39	19.36	0.9796
EMD-GAT-MGATCN	19.87	28.2	18.31	0.9777
EEMD-GAT-MGATCN	17.05	22.25	16.1	0.9879
CEEMD-GAT-MGATCN	17.87	22.61	16.95	0.9823

Table 4. Evaluation indexes of the compared model.

GAT- MGTCN, and CEEMD-GAT-MGTCN, the VMD-GAT-MGTCN with prediction values more fitting the actual traffic flow values obtains the better prediction results.

Furthermore, the evaluation indexes MAE, RMSE, MAPE, and R² are calculated and shown in Table 4. According to Tables 3 and 4, it can be found that, compared with GAT-MGATCN, the evaluation indexes of EMD-GAT-MGATCN, EEMD-GAT-MGATCN, CEEMD-GAT-MGATCN, and VMD-GAT-MGATCN have been reduced in different degrees. This is because the traffic flow time series data before decomposition is nonlinear and prone to mutation, while after decomposition through EMD, EEMD, CEEMD and VMD, the traffic flow modal components are obtained which more stable and tend to be linear, thus enabling the prediction model to more accurately obtain the temporal features and improve the accuracy of prediction. In addition, according to Tables 3 and 4, it can also be found that all evaluation indexes of VMD-GAT-MGTCN are the smallest. This is because the VMD algorithm effectively solves mode aliasing and end effect problem in the other three signal decomposition algorithms, while reducing the instability of traffic flow data, thus resulting in improved prediction performance. Although R² of EEMD-GAT-MGATCN is better than VMD-GAT-MGATCN, MAE, RMSE, and MAPE of EEMD-GAT-MGATCN are all larger than VMD-GAT-MGATCN, indicating that although EEMD-GAT-MGATCN can better capture the overall trend, the deviation of specific values is large. In general, VMD-GAT-MGATCN is better.

Model	Number of hidden neurons
LSTM	64
GRU	128
GAT	32
BiGRU	64

Table 5. Baseline models hyperparameter settings.

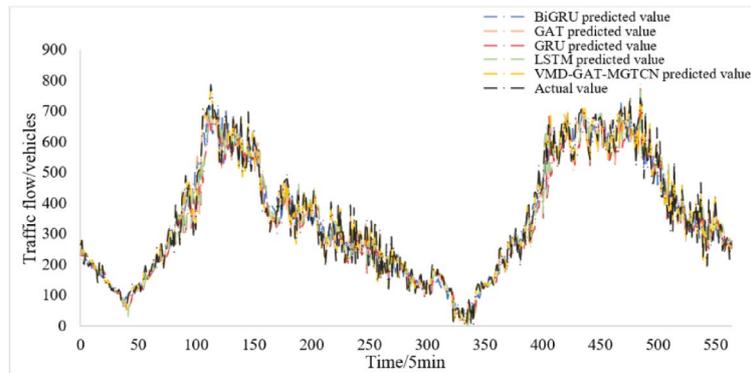


Fig. 10. Prediction results of the baseline model.

Experiments and analysis between VMD-GAT-MGTCN and baseline models

Four commonly used deep learning traffic flow prediction models, namely LSTM model, GRU model, GAT model, and BiGRU model, are selected as the baseline models.

(1) *Baseline model parameter settings* The initial learning rate, number of training rounds, loss function, and training batch size of the baseline models are the same as those of the VMD-GAT-MGTCN, and the rest of the hyperparameter settings are shown in Table 5.

(2) *Results and analysis* According to the parameter settings, the baseline models LSTM, GRU, GAT, and BiGRU are trained and tested under the same dataset, and the results are obtained and displayed in Fig. 10. For comparison, the prediction results of the VMD-GAT-MGTCN are also included in Fig. 10.

Figure 10 illustrates that both the VMD-GAT-MGTCN and the baseline models maintain a good overlap with the actual traffic flow values, and better prediction results are achieved. In order to observe the differences in the prediction results of each model more clearly, the two consecutive hours' results are randomly selected for two hours and are shown in Fig. 11. Based on Fig. 11, it is found that compared with the four baseline models, the VMD-GAT-MGTCN has a higher degree of overlap with the actual values of the traffic flow. This indicates the prediction results of the VMD-GAT-MGTCN are more aligned with actual traffic flow values and obtain better prediction effect.

Furthermore, the evaluation indexes MAE, RMSE, MAPE, and R^2 are calculated and shown in Table 6. According to Tables 3 and 6, compared with the four baseline models, the VMD-GAT-MGTCN has the smallest values of the evaluation indexes. This is because the VMD-GAT-MGTCN can better acquire the nonlinearity and instability features in traffic flow data and overcome the shortcomings of the baseline models that only acquires single feature, which results in better prediction accuracy than the baseline models.

(3) *Analysis of prediction results of traffic flow mutation region* Whether the traffic flow prediction method can obtain high accuracy in the traffic flow mutation region is an important reflection of prediction performance. To confirm the prediction capability of the VMD-GAT-MGTCN in the traffic flow mutation region, four traffic flow mutation regions are selected from Fig. 10, namely 90–100, 110–120, 250–260, and 485–495, which are numbered in chronological order as mutation region I, mutation region II, mutation region III, and mutation region IV.

According to the experiment, the prediction results of the VMD-GAT-MGTCN and the four baseline models in the four traffic flow mutation regions are obtained, which are illustrated in Fig. 12. Furthermore, MAE, RMSE, and MAPE are calculated and presented in Table 7.

Figure 12 indicates that compared to the four baseline models, the prediction results of the VMD-GAT-MGTCN exhibit a higher degree of overlap with the actual values. And Table 7 indicates that the evaluation indexes for the VMD-GAT-MGTCN are smallest in the four mutation regions. This indicates that the prediction effect of the VMD-GAT-MGTCN outperforms the baseline models. The reason is that in the traffic flow

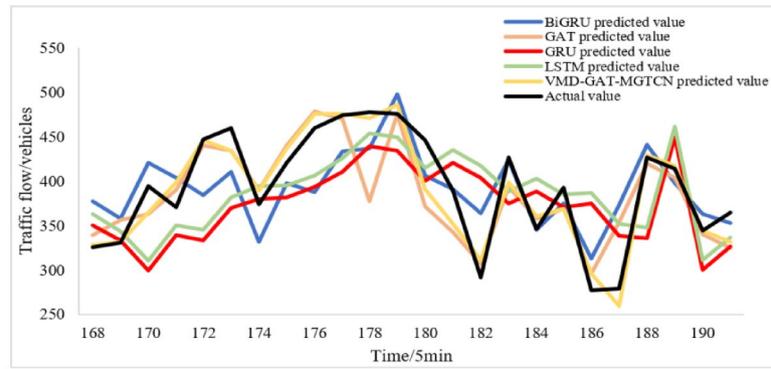


Fig. 11. Prediction results of the baseline model for two hours.

Model	MAE	RMSE	MAPE (%)	R ²
BiGRU	24.55	31.6	30.66	0.9549
GAT	23.41	30.11	14.63	0.9720
GRU	36.69	41.41	33.8	0.9066
LSTM	35.6	39.98	32.7	0.9060

Table 6. Evaluation indexes of the baseline models.

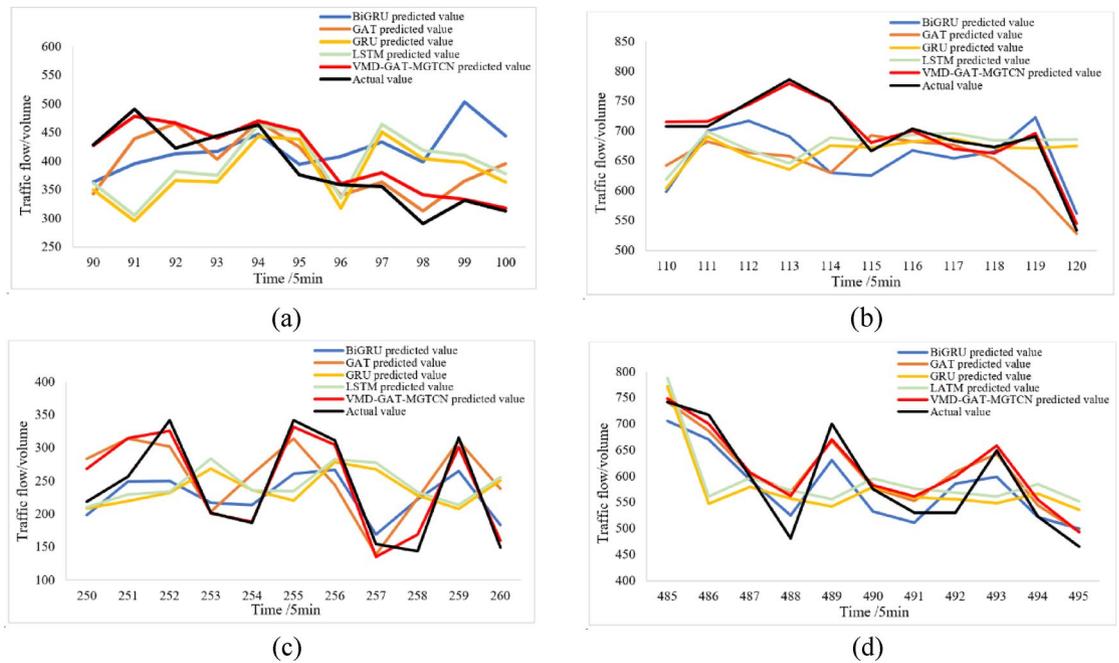


Fig. 12. Prediction results of five models in mutation regions. (a) Mutation region I, (b) Mutation region II, (c) Mutation region III, (d) Mutation region IV.

mutation regions, the baseline models fail to predict such mutations of the traffic flow and continue to maintain the previous trend, thus showing the opposite trend to the actual value, and even if the traffic flow mutation is predicted, the error is large, resulting in a decrease of the prediction accuracy. However, the VMD-GAT-MGTCN overcomes the above drawbacks and improves the prediction accuracy in the mutation region.

Conclusion and future work

A short-term traffic flow combined prediction model is proposed, VMD-GAT-MGTCN, which is based on VMD, GCN, attention mechanism, and TCN. The combined prediction model mainly consists of VMD and spatio-temporal feature model. The role of VMD is to decompose the traffic flow data into IMFs and reduce

Mutation region	Model	MAE	RMSE	MAPE (%)
Mutant region I	BiGRU	31.55	41.61	31.66
	GAT	23.91	33.19	14.70
	GRU	46.29	61.34	33.79
	LSTM	45.60	59.98	35.70
	VMD-GAT-MGTCN	15.59	23.65	11.75
Mutant region II	BiGRU	60.43	74.18	10.45
	GAT	59.60	73.22	10.17
	GRU	104.79	120.04	16.67
	LSTM	92.70	108.70	14.90
	VMD-GAT-MGTCN	5.56	6.55	0.92
Mutant region III	BiGRU	34.35	43.09	13.77
	GAT	33.88	47.54	14.58
	GRU	48.80	63.96	19.02
	LSTM	49.42	63.40	19.46
	VMD-GAT-MGTCN	12.46	15.68	5.04
Mutant region IV	BiGRU	36.94	41.87	6.22
	GAT	28.86	40.43	5.36
	GRU	66.36	85.18	10.89
	LSTM	71.14	84.63	11.96
	VMD-GAT-MGTCN	24.29	31.28	4.52

Table 7. Evaluation indexes in the baseline traffic flow mutation regions.

its instability. The spatio-temporal feature model is composed of GAT and MGTCN. The GAT is derived from adding multiple attention mechanisms in the GCN and used to capture spatial features. The MGTCN is composed of TCN and gating mechanisms and used to extract temporal features.

In order to verify the performance of the VMD-GAT-MGTCN, the PeMS04 dataset is selected, and the experiments are designed. The experimental results indicate that the VMD-GAT-MGTCN employing VMD for decomposing the traffic flow data and considering the spatio-temporal characteristics of traffic flow data achieved excellent predictive performance. Furthermore, the compared models GAT-MGTCN, EMD-GAT-MGTCN, EEMD-GAT-MGTCN, and CEEMD-GAT-MGTCN are designed and the effective function of the VMD on the prediction performance of the VMD-GAT-MGTCN is verified. In addition, compared with baseline models the VMD-GAT-MGTCN has superior prediction accuracy, and particularly in the traffic flow mutation region also obtains better prediction results.

The proposed model in this paper integrates the temporal and spatial features of traffic flow and introduces VMD to reduce the instability of traffic flow data. Simulation experiments demonstrate that good traffic flow prediction results are achieved by the proposed model. However, the actual traffic flow data often contains not only temporal and spatial features but also other factors such as cycle time, environment, and vehicle speed, etc. Therefore, considering more factors in traffic flow prediction should be further researched.

Data availability

The datasets generated and/or analysed during the current study are available upon request by contact with the corresponding author.

Received: 29 September 2024; Accepted: 11 April 2025

Published online: 17 May 2025

References

1. Wang, X., Zhang, F., Li, B. & Gao, J. Developmental pattern and international Cooperation on intelligent transport system in China. *Case Stud. Transp. Policy*. **5**, 38–44 (2017).
2. Liu, Y., Liu, J. & Dou, H. Effects of carbon emission on the environment of high-speed vehicles on highways for intelligent transportation systems. *Mobile Information Systems* 1–8 (2022). (2022).
3. Sarooraj, B., Prayla Shyry, S. & R. & Optimal routing with spatial-temporal dependencies for traffic flow control in intelligent transportation systems. *Intell. Autom. Soft Comput.* **36**, 2071–2084 (2023).
4. Alghamdi, T., Elgazzar, K., Bayoumi, M., Sharaf, T. & Shah, S. Forecasting traffic congestion using ARIMA modeling. In *International Conference on Wireless Communications and Mobile Computing* (2019).
5. Han, S. Y. et al. Adaptation scheduling for urban traffic lights via FNT-based prediction of traffic flow. *Electronics* **11**, 658–658 (2022).
6. Chen, W., Liu, B., Han, W., Li, G. & Song, B. Dynamic path planning based on traffic flow prediction and traffic light status. *Algorithms and Architectures for Parallel Processing*, ICA3PP PT I, 14487, (2024). (2023).
7. Okutani, I. & Stephanedes, Y. J. Dynamic prediction of traffic volume through Kalman filtering theory. *Transp. Res. Part. B: Methodological*. **18**, 1–11 (1984).
8. Cai, L. et al. A noise-immune Kalman filter for short-term traffic flow forecasting. *Phys. A: Stat. Mech. Its Appl.* **536**, 122601 (2019).

9. Isravel, D. P., Silas, S. & Rajsingh, E. B. Long-term traffic flow prediction using multivariate SSA forecasting in SDN based networks. *Pervasive Mob. Comput.* **83**, 101590 (2022).
10. Habtemichael, F. G. & Cetin, M. Short-term traffic flow rate forecasting based on identifying similar traffic patterns. *Transp. Res. Part. C: Emerg. Technol.* **66**, 61–78 (2016).
11. Chen, L., Zheng, L., Yang, J., Xia, D. & Liu, W. Short-term traffic flow prediction: from the perspective of traffic flow decomposition. *Neurocomputing* **413**, 444–456 (2020).
12. Yan, H. et al. Learning a robust classifier for short-term traffic state prediction. *Knowl. Based Syst.* **242**, 108368 (2022).
13. Yang, B., Sun, S., Li, J., Lin, X. & Tian, Y. Traffic flow prediction using LSTM with feature enhancement. *Neurocomputing* **332**, 320–327 (2019).
14. Zhang, R. et al. Short-term traffic flow forecasting model based on GA-TCN. *Journal of Advanced Transportation* (2021). (2021).
15. Wang, H., Zhang, R., Cheng, X. & Yang, L. Hierarchical traffic flow prediction based on Spatial-Temporal graph convolutional network. *IEEE Trans. Intell. Transp. Syst.* **23**, 16137–16147 (2022).
16. Gao, Y., Liu, Z., Du, J. & Wang, J. Spatial-temporal traffic flow prediction model based on the GAT and BiGRU. *J. Phys. Conf. Ser.* **2589**, 012024–012024 (2023).
17. Huang, H., Chen, J., Huo, X., Qiao, Y. & Ma, L. Effect of multi-scale decomposition on performance of neural networks in short-term traffic flow prediction. *IEEE Access.* **9**, 50994–51004 (2021).
18. Erdem, D. Robust-LSTM: a novel approach to short-traffic flow prediction based on signal decomposition. *Soft. Comput.* **26**, 5227–5239 (2022).
19. Wang, Y., Zhao, L., Li, S., Wen, X. & Xiong, Y. Short term traffic flow prediction of urban road using time varying filtering based empirical mode decomposition. *Appl. Sci.* **10**, 2038–2038 (2020).
20. Xin, C. et al. Traffic flow prediction at varied time scales via ensemble empirical mode decomposition and artificial neural network. *Sustainability* **12**, 3678–3678 (2020).
21. Lu, W., Rui, Y., Yi, Z., Ran, B. & Gu, Y. A hybrid model for lane-level traffic flow forecasting based on complete ensemble empirical mode decomposition and extreme gradient boosting. *IEEE Access.* **8**, 42042–42054 (2020).
22. NE, H. et al. T., The Empirical Mode Decomposition and the Hilbert Spectrum for Nonlinear and Non-Stationary Time Series Analysis. *Proc. R. Soc. Lond. A.* **454**, 903–995 (1998).
23. Zhaohua, W. & Norden, E. H. Ensemble empirical mode decomposition: A Noise-Assisted data analysis method. *Advances in adaptive data analysis.* **01**, 1–41 (2009).
24. Jia-Rong, Y., Jiann-Shing, S. & Norden, E. Complementary ensemble empirical mode decomposition: A novel noise enhanced data analysis method. *Adv. Data Sci. Adapt. Anal.* **02**, 135–156 (2010).
25. Konstantin, D. & Dominique, Z. Variational mode decomposition. *IEEE Trans. Signal Process.* **62** (3), 531–544 (2014).
26. Haichao, H., Jingya, C., Xinting, H., Yufei, Q. & Lei, M. Effect of Multi-Scale decomposition on performance of neural networks in Short-Term traffic flow prediction. *IEEE Access.* **9**, 50994–51004 (2021).
27. Emami, A., Sarvi, M. & Asadi Bagloee, S. Using Kalman filter algorithm for short-term traffic flow prediction in a connected vehicle environment. *J. Mod. Transp.* **27**, 222–232 (2019).
28. Wang, T., Tao, X., Zhang, J. & Li, Y. Predicting traffic congestion time based on Kalman filter algorithm. *Adv. Res. Reviews* **1**, 1–10 (2020).
29. Levin, M. & Tsao, Y. D. On forecasting freeway occupancies and volumes (abridgment). *Transportation Research Record* (1980).
30. Mai, T., Ghosh, B. & Wilson, S. Short-term traffic-flow forecasting with auto-regressive moving average models. *Proc. Institution Civil Engineers-transport.* **167**, 232–239 (2014).
31. Shahriari, S., Ghasri, M., Sisson, S. A. & Rashidi, T. Ensemble of ARIMA: combining parametric and bootstrapping technique for traffic flow prediction. *Transportmetrica A: Transp. Sci.* **16**, 1552–1573 (2020).
32. Chen, L. W. & Chen, D. E. Exploring Spatiotemporal mobilities of highway traffic flows for precise travel time Estimation and prediction based on electronic toll collection data. *Veh. Commun.* **30**, 100356 (2021).
33. Xu, D. et al. Real-time road traffic state prediction based on kernel-KNN. *Transportmetrica A: Transp. Sci.* **16**, 104–111 (2020).
34. Wang, Z., Ji, S. & Yu, B. Short-term traffic volume forecasting with asymmetric loss based on enhanced KNN method. *Math. Probl. Eng.* **2019**, 1–11 (2019).
35. Wang, D. et al. Bayesian optimization of support vector machine for regression prediction of short-term traffic flow. *IDA* **23**, 481–497 (2019).
36. Hu, J., Gao, P., Yao, Y. & Xie, X. Traffic flow forecasting with particle swarm optimization and support vector regression. In *International Conference on Intelligent Transportation Systems* 2267–2268 (2014).
37. Su, H., Zhang, L. & Yu, S. Short-term traffic flow prediction based on incremental support vector regression. In *Computing and Networking - Across Practical Development and Theoretical Research* (2007).
38. Zhu, Z., Peng, B., Xiong, C. & Zhang, L. Short-term traffic flow prediction with linear conditional Gaussian bayesian network. *J. Adv. Transp.* **50**, 1111–1123 (2016).
39. Sun, B., Cheng, W., Goswami, P. & Bai, G. Short-term traffic forecasting using self-adjusting k-nearest neighbours. *IET Intel. Transport Syst.* **12**, 41–48 (2018).
40. Ryu, U., Wang, J., Kim, T., Kwak, S. I. & U, J. Construction of traffic state vector using mutual information for short-term traffic flow prediction. *Transp. Res. Part. C: Emerg. Technol.* **96**, 55–71 (2018).
41. Lin, G., Lin, A. & Gu, D. Using support vector regression and K-nearest neighbors for short-term traffic flow prediction based on maximal information coefficient. *Inf. Sci.* **608**, 517–531 (2022).
42. Thomas, N. K. & Welling, M. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations* (2017).
43. Koesdwiady, A., Soua, R. & Karray, F. Improving traffic flow prediction with weather information in connected cars: A deep learning approach. *IEEE Trans. Veh. Technol.* **65**, 9508–9517 (2016).
44. Shu, W., Cai, K. & Xiong, N. N. A short-term traffic flow prediction model based on an improved gate recurrent unit neural network. *IEEE Trans. Intell. Transp. Syst.* **23**, 16654–16665 (2022).
45. Tian, Y., Zhang, K., Li, J., Lin, X. & Yang, B. LSTM-based traffic flow prediction with missing data. *Neurocomputing* **318**, 297–305 (2018).
46. Li, W., Wang, X., Zhang, Y. & Wu, Q. Traffic flow prediction over multi-sensor data correlation with graph Convolution network. *Neurocomputing* **427**, 50–63 (2021).
47. Chen, K. et al. Dynamic spatio-temporal graph-based CNNs for traffic flow prediction. *IEEE Access.* **8**, 185136–185145 (2020).
48. Qiao, Y., Wang, Y., Ma, C. & Yang, J. Short-term traffic flow prediction based on 1DCNN-LSTM neural network structure. *Mod. Phys. Lett. B* **35**, 2150042–2150042 (2021).
49. Sun, P., Boukerche, A., Tao, Y. & SSGRU A novel hybrid stacked GRU-based traffic volume prediction approach in a road network. *Comput. Commun.* **160**, 502–511 (2020).
50. Peng, H. et al. Dynamic graph convolutional network for long-term traffic flow prediction with reinforcement learning. *Inf. Sci.* **578**, 401–416 (2021).
51. Hu, N., Zhang, D., Liang, W., Li, K. C. & Castiglione, A. DSTGCS: an intelligent dynamic spatial-temporal graph convolutional system for traffic flow prediction in ITS. *Soft. Comput.* **28**, 6909–6922 (2024).
52. Yu, X., Sun, L., Yan, Y. & Liu, G. A short-term traffic flow prediction method based on spatial-temporal correlation using edge computing. *Comput. Electr. Eng.* **93**, 107219 (2021).

53. Liu, Q. et al. Explanatory prediction of traffic congestion propagation mode: A self-attention based approach. *Phys. A-statistical Mech. Its Appl.* **573**, 125940–125940 (2021).
54. Wang, K. et al. A hybrid deep learning model with 1DCNN-LSTM-attention networks for short-term traffic flow prediction. *Phys. A-statistical Mech. Its Appl.* **583**, 126293–126293 (2021).
55. Zhang, Q., Yin, C., Chen, Y. & Su, F. I. G. C. R. R. N. Improved graph Convolution res-recurrent network for spatio-temporal dependence capturing and traffic flow prediction. *Eng. Appl. Artif. Intell.* **114**, 105179 (2022).
56. Liu, H., Zhang, X., Yang, Y., Li, Y. & Yu, C. Hourly traffic flow forecasting using a new hybrid modelling method. *J. Cent. South. Univ.* **29** (4), 1389–1402 (2022).
57. Ke, Z., Dudu, G., Miao, S., Chenao, Z. & Hongbo, S. Short-Term traffic flow prediction based on VMD and IDBO-LSTM. *IEEE Access.* **11**, 97072–97088 (2023).
58. Lu, J. An efficient and intelligent traffic flow prediction method based on LSTM and variational modal decomposition. *Measurement: Sens.* **28**, 100843 (2023).
59. Yu, Y., Shang, Q. & Xie, T. A hybrid model for short-term traffic flow prediction based on variational mode decomposition, wavelet threshold denoising, and long short-term memory neural network. *Complexity* **2021** (7756299), 1–24. 7756299 (2021).
60. Bruna, J., Zaremba, W., Szlam, A. & LeCun, Y. Spectral networks and locally connected networks on graphs. In *International Conference on Learning Representations* (2013).
61. Mnih, V., Heess, N., Graves, A. & Kavukcuoglu, K. Recurrent models of visual attention. *Advances in neural information processing systems* **27**, (2014).
62. Bai, S., Kolter, J. Z. & Koltun, V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. ArXiv Preprint [arXiv:1803.01271](https://arxiv.org/abs/1803.01271) (2018).
63. Lu, B., Gan, X., Jin, H., Fu, L. & Zhang, H. Spatiotemporal adaptive gated graph convolution network for urban traffic flow forecasting. In *International Conference on Information and Knowledge Management* 1025–1034 (2020).

Author contributions

C.R., F.F. and C.Y. conceived and designed the experiments. F.F. and L.L. analysis the results. F.F. wrote the first draft of the manuscript. L.C. prepared Figs. 1, 2, 3, 4 and 5. All authors contributed to the manuscript and reviewed the manuscript.

Funding

Key Research and Development Project of Shandong Province (No. 2019GGX101008).

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to C.Y.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2025