ADVANCES IN TRANSPORTATION STUDIES
An International Journal

Editor in Chief: Alessandro Calvi

Vol. LXIX July 2026

Contents

aracne

Advances in Transportation Studies: an international Journal 69 (2026)

ISBN 979-12-218-2735-4 – ISSN 1824-5463 – DOI 10.53136/97912218273541 – pag. 3-26

Highway traffic state prediction method based on spatial-temporal correlation and physical information constraints

W. Feng W. Wu Y. Guo* Y. Gao W. Shan

School of Civil Engineering and Transportation-Northeast Forestry University, 26 Hexing Road, 150040, Harbin, China
**corresponding author; email: ying-guo@nefu.edu.com*

subm. 6th August 2025 approv. after rev. 3rd December 2025

Abstract

Highway traffic state prediction is needed for intelligent traffic control and decision-making. However, current approaches face challenges in modeling spatiotemporal features, addressing data sparsity, and maintaining adherence to physical laws. This study introduces a novel traffic state prediction model that combines physical information with a hybrid Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) network approach (PI-CNN-LSTM) to achieve precise road traffic state forecasting. Initially, a hybrid CNN-LSTM deep-learning architecture was developed to capture spatial-temporal features. The CNN was used to extract the spatial topology correlation features of the road network, and the LSTM was combined to model the time series evolution law of traffic parameters, and the temporal and spatial dependence of traffic states was captured synchronously. Subsequently, to ensure the predictions align with traffic flow conservation laws, a physical regularization loss function was formulated using the Greenberg fundamental diagram and Lighthill Whitham Richards (LWR) conservation equation. The fundamental diagram parameters, including free-flow velocity and blockage density, were determined by the neural network within the loss function. Finally, the model's generalization capability under sparse data was enhanced by training on subsets of the Canadian Whitemud Drive dataset at sampling rates of 10%, 20%, and 30%. The experimental results showed that on working days, when the data sampling rate was 10%, the Mean Square Error (MSE) of PI-CNN-LSTM was 13.24, the Mean Absolute Error (MAE) was 7.21, and the Mean Absolute Percentage Error (MAPE) was 6.98%. When the sampling rate was increased to 30%, the MSE was 12.05, which decreased by only 1.19 compared to the sampling rate of 10%. This indicates that the physical constraints significantly enhanced the model's ability to adapt to the data sparsity. In the highly volatile traffic scenario on non-working days, PI-CNN-LSTM maintained a stable performance. The MSE was 10.75 at a 10% sampling rate and dropped to 9.01 at a 30% sampling rate. Among all the evaluation indicators, compared with the CNN-LSTM and PIDL models, the error was significantly reduced, which proves the validity and superiority of the proposed model.

Keywords – traffic state prediction, physics-informed deep-learning, intelligent transportation systems, velocity prediction

1. Introduction

The significance of highways in transportation networks has grown with urbanization. Intelligent Transportation Systems (ITS) improve safety, efficiency, and sustainability. Traffic State Prediction (TSP) within ITS aids in decision-making. Sparse data from current highway monitoring

systems poses a challenge for accurate road section state prediction. Mining spatial-temporal correlation features is crucial for effective traffic management.

Existing traffic state prediction methods are mainly divided into two categories: model-driven and data-driven. Traditional model-driven approaches include Lighthill-Whitham-Richards (LWR) [1], Aw-Rasle-Zhang (ARZ) [2] and Payne-Whitham (PW) [3] traffic flow models. Although the model-driven method has strong physical interpretability, it is based on idealized assumptions and difficult to accurately describe the nonlinear dynamic behavior in actual traffic. Data-driven methods mainly use statistical models and machine learning. Statistical models include Markov model[4], infinite Gaussian mixture models [5], and support vector machines (SVM) [6]. Machine learning methods include k-nearest neighbors [7], Bayesian network models [8], and neural networks. In recent years, neural networks have become a research hotspot. Traditional neural network models include graph convolutional networks (GCN) [9], generative adversarial networks (GAN) [10], and graph neural networks (GAN) [11]. Although data-driven methods perform well in feature extraction and nonlinear fitting, they often ignore the intrinsic physical laws of the transportation system, resulting in insufficient generalization ability in scenarios where data is scarce or out of distribution. In recent years, Physical-informed Deep Learning (PIDL) [12] method tries to embed traffic flow theory into a deep learning framework, so as to balance data-driven flexibility and physical constraint consistency. However, existing physical PIDL methods have some limitations in spatio-temporal modeling capabilities. Moreover, the performance of the model is limited by the inherent simplification and idealized assumptions of the embedded physical equations, which cannot accurately describe the strong nonlinear characteristics of congestion areas, and also ignore the spatial heterogeneity and time variation of traffic parameters, which ultimately limit the prediction accuracy and generalization ability in real complex scenes. Therefore, the current PIDL method still has some limitations in terms of spatio-temporal modeling ability.

In order to accurately depict the transition of traffic states and capture the spatio-temporal correlation of traffic states, this paper proposes a physical information hybrid deep learning model (PI-CNN-LSTM) that integrates CNN-LSTM and LWR conservation equation. First, the Greenshields fundamental diagram and neural network forward and backpropagation estimated free-flow velocity and obstruction density. The designed loss function included the necessary parameters. Second, the nonlinear Greenberg fundamental diagram was used to recreate the LWR conservation equation and create a physical regularization loss function. Next, the CNN extracted the spatial correlation features of the road network topology, and the temporal evolution law of traffic parameters was modeled with LSTM to fully reflect the traffic states' spatial-temporal dependence. Finally, traffic state prediction was performed using the Canadian Whitemud Drive dataset and evaluated.

Its main contributions are summarized as follows:

1) A physics-based deep-learning method was proposed for traffic state prediction. Combining the classical traffic flow theory (LWR model based on Greenberg's fundamental diagram) with the deep-learning model (CNN-LSTM) improves the limitations of traditional approaches when modeling complex traffic states. By embedding the traffic flow model into a neural network framework, the complementary advantages of the theoretical model and data-driven method were achieved.

2) Comprehensively consider the spatial-temporal correlation of traffic status data. In the CNN-LSTM framework, CNN is utilized to extract spatial features, LSTM captures temporal dependencies, and combined with the physical constraints of the LWR model on traffic states, the

spatial-temporal correlations of traffic state data are comprehensively considered to improve prediction accuracy.

3) Compared with the existing data-driven prediction models, the model proposed in this paper effectively addresses the data sparsity problem in traffic state prediction through the collaborative modeling of physical priors and data-driven methods; this conclusion has been verified on sparse data with sampling rates of 10%, 20%, and 30%.

The remainder of this paper is organized as follows. Section 2 presents the literature review, and Section 3 describes the problem and model design. Section 4 describes the experimental design and results. Finally, Section 5 concludes the paper.

2. Literature review

2.1. Model-driven approaches

The physical properties of traffic flows are used to infer unobserved states in model-driven approaches. Traffic flow models like LWR and ARZ are used to anticipate the traffic state. Vishno [13] used the second-order Aw-Rascle-Zhang (ARZ) traffic model, connected car data, and fixed sensor data to anticipate traffic states with great precision. By combining the generalized AW-Rascle-Zhang model with two types of traffic flow characteristics, Zhao [14] compared three filtering approaches to real-time freeway traffic state estimation based on the first-order and second-order macroscopic traffic flow models using fixed and mobile data together data together. Compared with the first-order LWR model, the second-order model ARZ model can capture the transition from free flow to congestion state, so it can reflect the non-steady state characteristics of traffic flow more realistically. Other commonly used traffic state prediction methods are Kalman filter and particle filter, where Kalman filter achieves traffic state prediction by assimilating the selected flow model with the observations. For example, Chen [15] proposed a decision-level fusion model (SFEB-EKF) that combines the squared flow error boundary with the extended Kalman filter. This model is used to estimate the traffic state error boundaries of sections with and without traffic detectors on highways when the detector coverage is insufficient. Experiments show that this model can maintain an accuracy gap of less than 5% even when real samples are missing, providing a theoretical basis for the optimization of traffic detector layout. Particle filtering predicts traffic states by iteratively weighting and resampling particles to approximate the posterior probability distribution. Hawes [16] suggested using a particle filter with a macroscopic traffic model to predict traffic, minimize computing time, and enhance model accuracy. Model-driven techniques have fewer parameters, clear physical meanings, concise formulations, and good interpretability, but they are generated under idealized theoretical conditions with many and unduly tight simplifying assumptions. These models' predictions often differ greatly from real-world traffic situations.

2.2. Data-driven approaches

Data-driven approaches have gained popularity as data-acquisition tools and computational power improve. Xu [17] proposed a deep-learning traffic state prediction framework. The system predicts traffic state using neighboring information, graph embedding to represent the route, and a generative adversarial network. This model predicts better, according to research. Zhang [18] developed a data-driven optimization method to estimate traffic. ETC data were used instead of detector data. Traffic state prediction is improved by using the detector data's fine granularity and the ETC's vast coverage.

With continuous research on data-driven approaches, an increasing number of studies have begun to adopt combined model approaches for traffic state prediction. Lu [19] proposed a comprehensive short-term traffic flow prediction method based on an Autoregressive Integrated Moving Average model (ARIMA) and an LSTM neural network. These two technologies were combined to achieve short-term traffic-flow prediction. Ding [20] proposed a traffic state prediction method in a low-dynamic environment with a low penetration rate of Internet-connected vehicles based on a deep-learning algorithm. Research has demonstrated that the performance of this model is significantly improved.

To improve the accuracy of model predictions further, researchers have gradually explored models that can simultaneously capture the temporal and spatial characteristics of traffic flow data. Xu [21] proposed a GATs-GAN framework for traffic state prediction. In order to capture the spatial traffic relationship, a traffic topology map network based on traffic link connections is established. Then, the first-order and higher-order neighbors of the traffic network can be structured. Graph Attention Networks (GATs) are used to obtain the hidden features of input traffic data by training the attention between high-order neighbor nodes. Experimental results show that GATs-GAN has higher prediction accuracy. Liu [22] designed a new traffic flow prediction method based on an autoencoder, called Spatio-Temporal Autoencoder (ST-AE). The prediction is performed by simply projecting the current hidden state to the future hidden state and then using the trained autoencoder to reconstruct the future traffic flow. Jin [23] proposed a spatial-temporal graph neural point processing structure (STGNPP) for traffic congestion event prediction. Experiments have proven that it has superior performance compared to the other models.

Overall, data-driven models are versatile. Its portability and interpretability are limited by the lack of theoretical assumptions. Additionally, data-driven systems require a lot of previous data and excellent data quality to accurately anticipate. Actual data collection is minimal. If there is enough data and the distribution matches the state to be anticipated, the prediction accuracy may be higher than the model-driven method. When empirical data is few or the traffic condition to be forecasted is distributed differently from empirical data, the prediction effect may be poor.

2.3. PIDL approaches

To address the limitations of the existing methodologies, researchers have integrated theoretical traffic flow models from model-driven approaches with deep-learning models in data-driven approaches, introducing a novel strategy called PIDL. Shi [24] revised the PIDL framework to include traffic flow fundamental diagram relations and presented a physical information deep-learning based on the fundamental diagram learner. Pereira [25] proposed a physically aware recurrent neural network algorithm. The algorithm embeds the discretization of the macroscopic traffic flow model into the network architecture to generate traffic state predictions, as well as traffic flow and velocity predictions. Yuan [26] developed a hybrid framework that combined a machine learning (ML) approach and classical traffic flow models to encode traffic flows as multivariate Gaussian processes. Based on this framework, they introduced progressive physical regularization learning to encode a complex traffic flow model incrementally into the ML process. Thodi [27] introduced a deep convolutional neural network (deep CNN) based on moving waves to predict high-resolution traffic velocity fields from sparse detection vehicle trajectories. Zhang [28] proposed a TSE model that combined PIDL to solve the problem of traffic state prediction in data-sparse scenarios. Studies have shown that the accuracy of traffic-state prediction has improved. Pan [29] proposed a hybrid interpretable method that combined a fundamental diagram (FD), Markov chains, and long short-term memory (LSTM). Traffic conditions were predicted by integrating the statistical data in both congested and non-congested scenarios.

Through the analysis of the above literature, it can be seen that model-driven methods have strong physical interpretability. However, these models are derived under ideal theoretical conditions, with relatively many simplifying assumptions and too strict, and the model prediction performance usually has a large deviation from the real traffic state. In addition, the traffic flow model may not accurately simulate the dynamics of traffic flow over time, especially during peak hours and when accidents occur, and the regularity of traffic flow may be broken. As a result, there are some differences between the traffic flow model and the actual traffic behavior. Therefore, there are some limitations when using model-driven methods for traffic state prediction. The data-driven approach has a flexible model form, but its portability and interpretability are relatively poor due to the lack of theoretical assumptions. In addition, the prediction accuracy of data-driven methods largely depends on a large amount of historical data, and the quality of the data is very high, and only limited data can be obtained in reality. When the amount of data is sufficient and the distribution is the same as the state to be predicted, the prediction accuracy may exceed that of model-driven methods. However, when the amount of empirical data is limited or the distribution of the traffic state to be predicted is different from the empirical data, the prediction effect may be poor. PIDL, which effectively combines model-driven and data-driven methods, can use theoretical knowledge to guide the training of neural networks, so as to obtain higher prediction accuracy. However, most PIDL frameworks in previous studies fail to fully incorporate spatial-temporal deep learning architectures when combining physical equations, resulting in limited ability to capture complex spatial-temporal dynamics of transportation systems. Secondly, the embedded physical model itself has applicable boundaries. For example, the Greenshields basic graph often used with LWR model assumes a linear relationship, which is difficult to accurately describe the nonlinear flow-density relationship in congestion state, thus limiting the prediction accuracy of the model in real complex scenes. In addition, PIDL usually assumes that the physical model used can be applied globally, ignoring the spatial heterogeneity and time-varying characteristics of traffic flow characteristics in different regions and time periods of the actual road network. Therefore, the current PIDL method still has some limitations in terms of spatial-temporal modeling ability.

According to the preceding literature, combining transportation theory with deep-learning methodologies and employing the theory to train neural networks can improve prediction accuracy. The geographic-temporal correlation of traffic status data has not been completely explored in this research, and they cannot effectively extract the temporal and spatial properties of the data.

3. Problem description and model design

3.1. Problem description

Traffic data contains spatial and temporal characteristics, with the spatial characteristics mainly reflected by the implicit spatial topological structure of the data. The entire spatiotemporal domain $D = （X，T）$ corresponds to road section X and time period T, where X represents the time span and T represents the spatial span. As shown in Eq. (1)., the D definition is given in the form of a matrix.

$$D = \begin{bmatrix} (x_1, t_1) & \cdots & (x_1, t_j) \\ \vdots & \ddots & \vdots \\ (x_i, t_1) & \cdots & (x_i, t_j) \end{bmatrix} \tag{1}$$

where $i \in \{1，2，\ \ldots，I\}，\ j \in \{1，2，\ \ldots，J\}$.

(x_i, t_j) represent the position and time of each grid point, $v(x_i, t_j)$ represent the velocity of the corresponding points. The velocity of the entire road section is represented by the traffic state

observation matrix S which consists of $v_o(x_i, t_j)$, and the traffic state matrix mapped $\hat{S}$ in the entire spatiotemporal region is constituted by $v_s(x_i, t_j)$. The mask matrix $M(x_i, t_j)$ denotes the locations in the traffic state observation matrix that lack sampling information. When there are no sampled data, $M(x_i, t_j) = 0$. Otherwise, $M(x_i, t_j) = 1$. To characterize the uncertainty of the unobserved traffic state in the matrix, a random variable $Z \sim p(z)$ is introduced into the input data. This random variable was used to capture the uncertainty owing to data sparsity. Traffic state prediction determines the mapping function g from the observation matrix $\hat{S}$ containing partial traffic state information, to the complete traffic state prediction matrix. The traffic state prediction problem can be expressed as follows:

$$\tilde{S} = S * M(x_i, t_j) + Z * [1 - M(x_i, t_j)] \quad (2)$$

$$\hat{S} = g(\tilde{S}) \quad (3)$$

$\hat{S} = g(\tilde{S})$ indicates that the sparse observation matrix $\tilde{S}$ is modeled by the mapping function g to generate a complete traffic state prediction matrix $\hat{S}$. $\tilde{S} = S * M + Z * (1 - M)$ by separating the probability distribution of the observed and unobserved regions.

3.2. The overall architecture of the PI-CNN-LSTM model

The overall framework of the model consists of three parts, as shown in Fig 1.

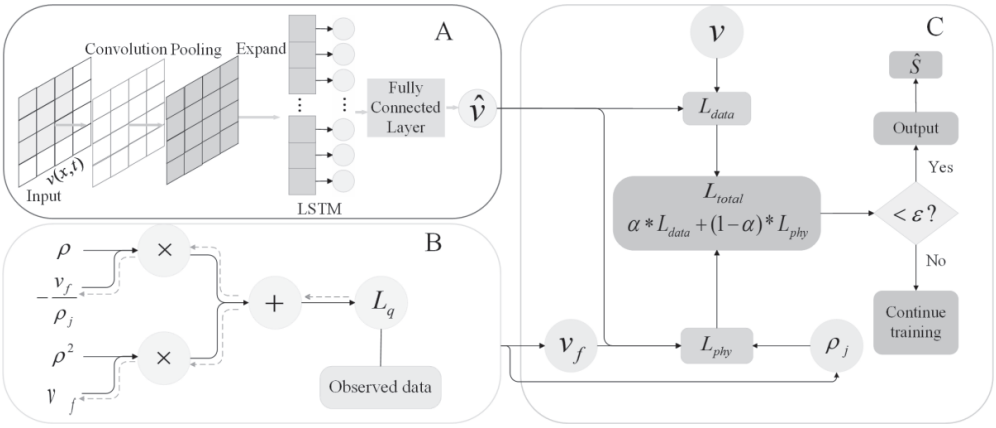


Fig. 1 - Overall architecture of the model

Part A discusses CNN-LSTM hybrid deep learning. Initial input data went through the Convolution Layer. After dimension-reduction processing by the Pooling Layer, the main feature subset is obtained. Finally, the LSTM network creates a deep feature representation of the extended time series. Velocity prediction values were used to calculate loss.

Part B presents the parameter determination procedure fundamental schematic. A computational graph shows the fundamental diagram parameter determination process, including each parameter and variable in the equation. Density and square term are input data. The neural network forward propagation estimated flow value was determined. In back-propagation, the mean square error between the anticipated flow and the true value is determined using the actual observed flow data as the supervisory signal. Finally, gradient descent calculates free flow velocity and obstruction density for the loss function computation.

Part C covers loss function design. The loss function is the weighted sum of the data difference and physical difference. The data differences are the difference between the predicted and actual values, while the physical differences come from the traffic flow model, which includes the fundamental diagram parameters of free-flow velocity and blockage density solved in Part B, and the data characteristics based on observation points and physical laws of traffic flow. In the process of model training, a dynamic stopping mechanism based on a relative convergence criterion is used. The optimizer stops training when it detects that the gradient or function value has changed less than a preset tolerance (tolerance_grad=1e-05, tolerance_change=machine epsilon) or when the maximum number of iterations is reached (10,000). This setting ensures that the model avoids overfitting while fully learning the physical laws, and has better adaptability on different problems. Otherwise, training proceeded. This model aims to achieve dynamic simulation and prediction of the entire traffic system state based on sparse real-time detection data. At the temporal level, the model's input is a continuous sequence of 12 time steps of historical data, and the prediction target is the complete traffic state matrix for the subsequent 3 time steps.

3.3. CNN-LSTM hybrid deep-learning model

The traffic-state prediction process comprises two primary stages: CNN and LSTM. Initially, the CNN analyzes the velocity data through convolution and pooling operations to identify spatially localized features. The output of the CNN is a "spatial feature vector" extracted from the traffic data at each time instant. The vector is the spatial pattern encoding of the traffic state of the road section after the original data is convoluted and pooled. These features can reflect the interaction between different intersections, such as the flow pattern of traffic flow and the propagation of traffic congestion. Subsequently, an LSTM network is employed to model the temporal progression of the spatial features.

3.3.1. The stage of extracting spatial features of traffic state

The spatial feature extraction of traffic states aims to extract and analyze the key features of traffic states from traffic data. Fig 2 shows the feature extraction process of the CNN, which consists of a convolution operation and a subsampling function. The calculation formula for traffic feature convolution and activation is as follows:

$$h_i^{ac} = f\left(\sum_{i\in F} \tilde{S}\otimes C_i\right) \tag{4}$$

In the formula, h_i^{ac} is the traffic feature vector output using the i-th convolution kernel. The f is the activation function used to enhance the ability of the convolution kernel to extract nonlinear traffic features.

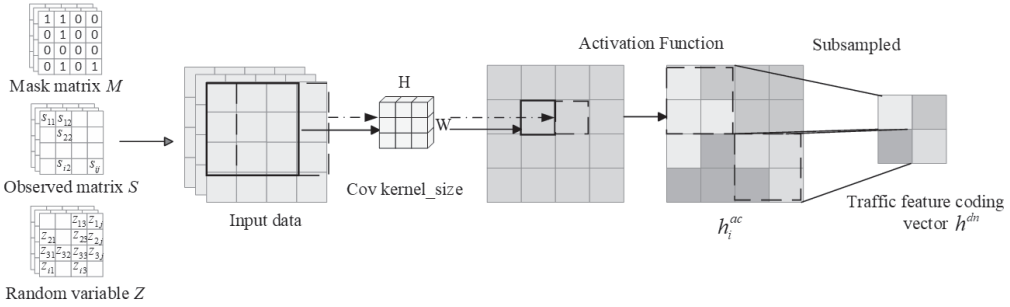


Fig. 2 - Feature extraction process of the CNN network

The activated traffic feature map h_i^{ac} generates the traffic feature vector h^{dn} through a subsampling operation. The h^{dn} serves as the input of the LSTM network. The subsampling operation of the traffic characteristics is expressed as follows:

$$h_i^{dn} = max\left(h_i^{ac} * u(p,q)\right) \tag{5}$$

where $u(p,q)$ is a two-dimensional sliding window for subsampling traffic features and (p,q) represent the size of the window.

3.3.2. The stage of extracting the temporal characteristics of traffic state

The LSTM layer uses the traffic state variable output of the CNN layer to represent the predicted values that are recursively generated for future time steps. The structure of the LSTM network is shown in Fig 3. The LSTM layer accepts the final traffic state vector (h_t, C_t) of the CNN layer as the initial hidden state and generates the prediction of future time steps through LSTM.

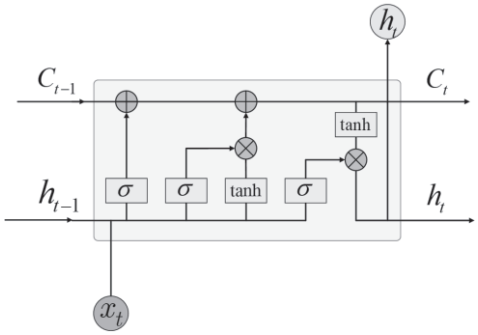


Fig. 3 - LSTM network structure

Finally, the values output by LSTM were used for the subsequent calculation of the loss function.

3.4. Fundamental diagram parameter determination

Before conducting traffic state prediction, it is necessary to determine the parameters of the fundamental diagram, including the free flow velocity v_f and the blockage density ρ_j. In order to deal with the problem of data sparsity, the estimation results of traditional statistical methods are unstable when the data is limited, while neural network can learn more robust parameters from sparse data through its powerful fitting ability and gradient optimization. This study adopted a method that combines a theoretical framework based on the Greenshield fundamental diagram with neural networks to determine the parameters. By combining traffic flow theory and neural network, the determination of basic graph parameters makes the whole model retain high interpretability. Using the basic theory of the physical model as prior knowledge makes it easier for researchers and practical users to understand the output of the model and the logic behind it. Therefore, the method of combining the theoretical framework based on Greenshields basic graph with neural network not only ensures the physical meaning of the parameters, but also improves the performance and practicability of the overall model under sparse data.

3.4.1. The theoretical basis of traffic flow

(1) The basic traffic flow equation is one of the most fundamental and important equations in traffic flow theory, as shown in Eq. (6):

$$q = \rho * v \tag{6}$$

where q represents the number of vehicles passing through within a unit of time, ρ represents the number of vehicles existing on a unit length of road within a given time, and v represents the average velocity of vehicles within a unit of time.

(2) The Greenshield fundamental diagram uses a linear function to determine the relationship between velocity and density, as shown in Eq. (7).

$$v = v_f(1 - \frac{\rho}{\rho_j}) \tag{7}$$

where v_f represents the free flow velocity, and ρ_j represents the blockage density.

According to Eq. (6) and Eq. (7), the relationship between the density and flow rate described by the Greenshield fundamental diagram can be determined as shown in Eq. (8). A density-flow diagram is shown in Fig. 4. The figure q_c represents the maximum traffic capacity, ρ_c denotes the critical density, which is the density when the flow reaches its peak, and ρ_j is the blockage density.

$$q = \rho v_f(1 - \frac{\rho}{\rho_j}) = \rho v_f - \frac{\rho^2 v_f}{\rho_j} \tag{8}$$

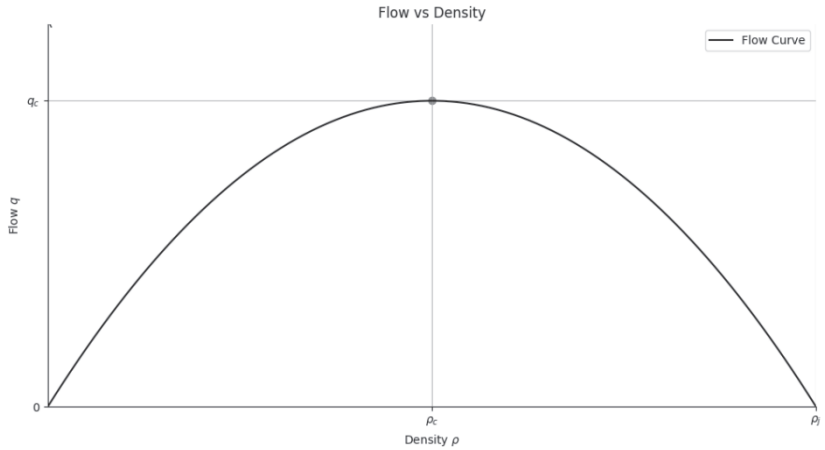


Fig. 4 - Density-flow plot of Greenberg's fundamental diagram

3.4.2. Neural network structure

To learn the free flow velocity v_f and the blockage density ρ_j from the data, a simple neural network including an input layer, a hidden layer, and an output layer is constructed. First, we reconstruct Eq. (8) in the form of a matrix operation that is more suitable for the matrices of neural networks, as shown in Eq. (9) :

$$q = \begin{bmatrix} \rho \\ \rho^2 \end{bmatrix} \cdot \begin{bmatrix} v_f \\ -\frac{v_f}{\rho_j} \end{bmatrix} \tag{9}$$

Secondly, in the forward propagation stage, the input X is the density and its square terms (ρ, ρ^2) which are respectively linearly transformed with the weight matrix, and the weight matrix directly corresponds to the physical parameters, as shown in Eq. (10). The predicted traffic value $\hat{q}$ is generated through the operation of a single-layer, fully connected layer, and its mathematical expression is given by Eq. (11).

$$W = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} v_f \\ -\frac{v_f}{\rho_j} \end{bmatrix} \tag{10}$$

$$\hat{q} = X \cdot W = \rho w_1 + \rho^2 w_2 \tag{11}$$

Finally, in the backpropagation stage, taking the actual observed flow data as the supervisory signal, the mean square error L_q between the predicted flow and the true value is calculated. The calculation formula of L_q is shown in Eq. (12):

$$L_q = \frac{1}{N}\sum_{i=1}^{N}(q_i - \hat{q}_i)^2 \tag{12}$$

where q_i is the true value, and $\hat{q}_i$ is the predicted value.

After calculating the mean square error L_q, the gradient descent method is used to dynamically update the weight parameters. Eq. (13) is the calculation formula for the gradient descent method.

$$w_1^{(k+1)} = w_1^{(k)} - \eta\frac{\partial L_q}{\partial w_1}, \quad w_2^{(k+1)} = w_2^{(k)} - \eta\frac{\partial L_q}{\partial w_2} \tag{13}$$

where k represents the number of iterations, and η represents the learning rate.

By using the gradient descent method, the values of w_1 and w_2 can be determined, and the parameters free flow velocity v_f and blockage density ρ_j in the Greenshields fundamental diagram can be further determined. After determining the parameter free-flow velocity and blocking density in the Greenshields fundamental graph, the loss function is constructed using the LWR and Greenberg fundamental graph. The strategy of separating parameter estimation from physical constraints is adopted in the framework, which can balance the model stability under the condition of sparse data and the description accuracy of complex traffic dynamics. Firstly, based on the simple structure and strong parameter identification of Greenshields model, a simple neural network was designed to estimate the two key parameters of free flow velocity and block density from sparse data. These parameters serve as prior knowledge and provide the basis for the subsequent construction of Greenberg's model. Then, the estimated parameters are substituted into the Greenberg model, which has more nonlinear representation ability, to construct the physical regularization loss function. The design not only realizes the dynamic calibration of physical constraints according to actual data, but also takes full advantage of Greenberg model in describing non-steady processes such as congestion formation and evolution. Through the synergy of the two, the PI-CNN-LSTM model can maintain high prediction accuracy and physical consistency in the presence of congestion, and improve the reliability and interpretability of prediction results.

3.5. Construction of loss function

The loss function was divided into two parts: data difference and physical difference. The data difference is the mean square error between the predicted values of the CNN-LSTM and the true

values. The physical differences were jointly constructed using the LWR traffic flow model and Greenberg fundamental diagram. The total loss function is the weighted sum of the two functions, and its specific expression is given by Eq. (14).

$$L_{total} = \alpha * L_{data} + \ （1 - \alpha）* L_{phy} \tag{14}$$

where L_{data} represents the data difference, L_{phy} represents the physical difference, and the hyperparameter $\alpha \in [0,1]$ is the weight used to balance data-driven and physical constraints.

The data difference is the mean square error between the predicted and true values of the CNN-LSTM. The calculation formula is given by Eq. (15).

$$L_{data} = \frac{1}{N_0}\sum_{i=1}^{N_0}\left|v\left(x_0^i, t_0^i\right) - \hat{v}\left(x_0^i, t_0^i\right)\right|^2 \tag{15}$$

The LWR model is a classic traffic-flow model based on the law of conservation. The basic assumption is that traffic flow is conserved; that is, the number of vehicles flowing from one end of the road is equal to the number of vehicles flowing out from the other end of the road. Eq. (16) provides a mathematical description of the LWR traffic flow model:

$$\frac{\partial \rho(x,t)}{\partial t} + \frac{\partial q(x,t)}{\partial x} = 0 \tag{16}$$

where $\rho(x,t)$represents the traffic flow density at position x and time t, and $q(x,t)$ represents the corresponding traffic flow.

The Greenberg fundamental diagram, as a nonlinear density-velocity model, is the mathematical expression shown in Eq. (17). The Greenberg fundamental diagram characterizes the relationship between velocity and density using a logarithmic function that is more consistent with the actual distribution of traffic data. In contrast, the Greenshields base plot, as a linear velocity-density relationship, applies to relatively stable traffic flows.

$$v = v_m ln\frac{\rho_j}{\rho} \tag{17}$$

where $v_m = {v_f}/{e}$ is the optimal velocity, and ρ_j is the blockage density.

To construct the physical differences of the LWR traffic flow model based on Greenberg and obtain $\frac{\partial \rho(x,t)}{\partial t}$, it is necessary to convert the relationship of density expressing velocity in Equation (17) into the relationship of velocity expressing density, as shown in Eq. (18) :

$$\rho = \rho_j e^{\frac{v}{v_m}} \tag{18}$$

By combining Eq. (6) and Eq. (18), the relationship between the flow rate q and velocity v can be obtained as follows:

$$q = \rho * v = \rho_j e^{\frac{v}{v_m}} * v \tag{19}$$

Taking the partial derivatives of Eq. (18) and (19) respectively, the derived $\frac{\partial \rho(x,t)}{\partial t}$and $\frac{\partial q(x,t)}{\partial x}$ are shown in Eq. (20) and (21) as follows:

$$\frac{\partial \rho(x,t)}{\partial t} = -\frac{\rho_j}{v_m} e^{\frac{v}{v_m}} \frac{\partial v(x,t)}{\partial t} \tag{20}$$

$$\frac{\partial q(x,t)}{\partial x} = (-\frac{\rho_j}{v_m} e^{\frac{v}{v_m}} + \rho_j e^{\frac{v}{v_m}}) \frac{\partial v(x,t)}{\partial x} \tag{21}$$

Substituting Eq. (20) and (21) into Eq. (16), the physical differences expressed by the LWR model combined with the fundamental diagram can be obtained as shown in Eq. (22). The

parameters in the physical differences, the optimal velocity v_m and the blockage density ρ_j, are determined using the method in Section 3.2. The predicted value $\hat{v}$ of the velocity obtained through CNN-LSTM is input into Eq. (22), to determine whether the prior point conforms to the physical constraints of the traffic flow model. The closer the obtained result is to zero, the more accurate is the prediction.

$$\begin{aligned} L_{phy} &= \frac{\partial \rho(x,t)}{\partial t} + \frac{\partial q(x,t)}{\partial x} \\ &= -\frac{\rho_j}{v_m} e^{\frac{v}{v_m}} \frac{\partial v(x,t)}{\partial t} + \left(-\frac{\rho_j}{v_m} e^{\frac{v}{v_m}} + \rho_j e^{\frac{v}{v_m}}\right) \frac{\partial v(x,t)}{\partial x} \end{aligned} \tag{22}$$

Substituting Eq. (22) and (15) into Eq. (14) yields the final loss function. As shown in Eq. (23), the final loss function consists of two parts: the data difference L_{data} and the physical difference L_{phy}. Both were measured using the mean square error. This difference lies in the fact that the data difference is the difference between the observed value and predicted value at the observation point. The physical difference is obtained by substituting the predicted value of the previous point into the physical loss function.

$$\begin{aligned} L_{total} &= \alpha * L_{data} + (1-\alpha) * L_{phy} \\ &= \frac{\alpha}{N_0} \sum_{i=1}^{N_0} \left| v\left(x_0^i, t_0^i\right) - \hat{v}\left(x_0^i, t_0^i\right) \right|^2 \\ &+ \frac{(1-\alpha)}{N_l} \sum_{j=1}^{N_l} \left| -\frac{\rho_j}{v_m} e^{\frac{\hat{v}\left(x_l^j, t_l^j\right)}{v_m}} \frac{2\hat{v}\left(x_l^j, t_l^j\right)}{\partial t} \right. \\ &+ \left. \left(-\frac{\rho_j}{v_m} e^{\frac{\hat{v}\left(x_l^j, t_l^j\right)}{v_m}} + \rho_j e^{\frac{\hat{v}\left(x_l^j, t_l^j\right)}{v_m}} \right) \frac{2\hat{v}\left(x_l^j, t_l^j\right)}{\partial x} \right|^2 \end{aligned} \tag{23}$$

where, $\left(x_0^i, t_0^i\right)$ is the spatial location and time of the observed data point, i and is the observed data point. x_l^j, t_l^j is the auxiliary space location and time used to compute the physical constraints, and j is the physical regularization computation point. x_0^i, t_0^i and x_l^j, t_l^j are input variables, i and j are independent of the other variables appearing in the previous text. N_0 is the total number of training data points with true observations. N_l represents the total number of auxiliary computation points used to compute the physical regularization term. L_{data} is calculated based on actual observed data and model predicted values. $\left(x_0^i, t_0^i\right)$ compares the mean squared error between the speed predicted $\hat{v}$ by the model and the true speed v at the observed point. L_{phy} does not rely on the actual observed data, but evaluates whether the predicted speed v of the model conforms to the preset physical law of traffic flow at the configuration points（x_l^j, t_l^j）. L_{phy} is not a fixed value and is one of the variables to be optimized during model training. It varies with the model parameters, since both the model prediction velocity $\hat{v}$ and its partial derivatives are functions of the model parameters.

4. Experimental design and result discussion

4.1. Introduction to the dataset

This study tested the strategy using a dataset from Canada's Whitemud Drive east-west route. This collection includes daily traffic flow, density, and velocity data from 9 ground induction coils for a 28-kilometer highway segment. Traffic status was characterized by velocity. The research area and velocity fluctuation graphs are in Fig 5. The data was taken from 15:00-17:00 on August 11 and 17, 2015.

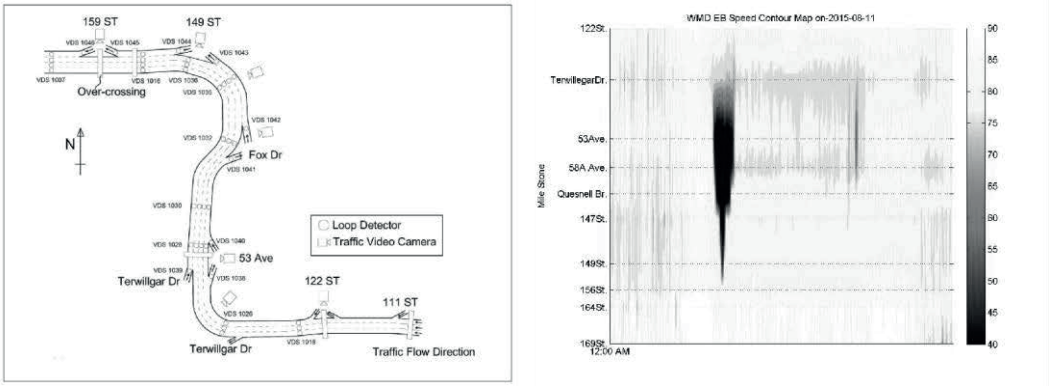


Fig. 5 - The study area and variation of velocity

(1) Create working and non-working day datasets. In order to verify the generalization ability of the model in different traffic modes, weekday and non-weekday data sets are constructed respectively. The prediction task in this study is defined as the prediction of a speed sequence over a future period of time using a historical speed sequence over a past period of time. Specifically, taking 5 minutes as a basic time step, historical data from 12 consecutive time steps (i.e., the past 1 hour) is used as input to predict the traffic state for the next 3 time steps (i.e., the next 15 minutes). Using the same model architecture, after training on the complete training set, the weekday and non-weekday test sets are independently predicted and evaluated.

(2) Prepare original data. First, wavelet-denoising is explained. Wavelet denoising was used to isolate signals from noise, extract dominating features, and reduce the impact of high-frequency noise on prediction outcomes. Second, wavelet-denoised data was Min-Max normalized. Min-Max normalization standardised the data to the [0,1] interval, reducing biases in the raw data.

(3) Different sampling rates for sparse data. In order to verify the ability of the model to learn from sparse data, when constructing the training set, sparse data with different sampling rates (10%, 20%, 30%) are randomly selected from the complete spatial-temporal domain as labeled samples, and the remaining large number of unobserved spatial-temporal points constitute the test set. This setup aims to simulate a data-scarce scenario in practical applications and quantify its generalization performance by the global error between the model prediction and the full true field. This method is widely used in the field of physical information neural networks, which can effectively prove that the model is not overfitting to a small amount of data, but really learns the internal physical laws, so as to achieve accurate prediction.

4.2. Spatial correlation analysis

In order to statistically verify the spatial dependence of traffic state data itself and provide an objective basis for the subsequent use of Convolutional Neural Network (CNN) for spatial feature extraction, this subsection analyzes the spatial autocorrelation of experimental data.

Spatial autocorrelation refers to the degree to which the values of a variable at a spatial location are correlated with the values of the same variable at its neighboring locations. Global Moran's I is used as the core measure in this study. Moran index is a classical spatial autocorrelation statistic. Its range is usually around 0, and its range is roughly between [-1, 1]. $I > 0$ indicates positive spatial correlation, and the closer its value is to 1, the stronger the spatial clustering is. $I < 0$ indicates negative spatial correlation. Then, $I \approx 0$ means that the spatial distribution is random and there is no autocorrelation. Table 1 shows the results of the descriptive statistics of Moran index for weekday and non-weekday data.

Tab. 1 - Global Moran's I descriptive statistics

Statistics	working day datasets	non-working day datasets
Mean value	0.648	0.621
Standard deviation	0.075	0.098
Minimum value	0.521	0.453
Maximum value	0.781	0.752
Median	0.655	0.630
Proportion with a p-value < 0.01	100%	100%

As shown in Table 1, the mean values of Moran index for weekday and non-weekday data are 0.648 and 0.621, respectively, and the medians are all above 0.63, and the Moran index on all time slices is positive. This clearly confirms statistically that there is a strong positive spatial correlation between the velocity values of each detector on a highway segment, that is, the velocity of a segment is highly correlated with the velocity of its neighboring segments, rather than being randomly distributed.

4.3. Experimental setup

The experimental setup was divided into two parts: determination of the fundamental diagram parameters and traffic state prediction. In the process of determining the fundamental diagram parameters, a fully connected neural network (FCNN) is used to identify v_f and $-\frac{v_f}{\rho_j}$. The FCNN consists of input, output, and hidden layers. The input data is (ρ, ρ^2), and the output is the traffic flow q. The neural network has only two learnable parameters: v_f and $-\frac{v_f}{\rho_j}$. Table 2 lists the parameter settings for the fully connected neural network.

Tab. 2 - Parameter settings of fully connected neural networks

Optimizer	Adam
Learning rate	0.01
Number of iterations	5000
Loss function	MSE

The traffic state prediction process was implemented using the PyTorch deep-learning framework. The entire framework is composed of CNN convolutional and LSTM layers. The input data are the spatial-temporal state information $v(x,t)$, and the output is the traffic feature map. Subsequently, The obtained traffic feature vectors were flattened into one-dimensional vectors and used as inputs for the LSTM layer. The model was executed on an Intel(R) Core(TM) i7 - 10750H CPU @ 2.60GHz. The parameter settings of the model are listed in Table 3.

Tab. 3 - Model parameter settings

CNN layer parameter setting	LSTM layer parameter setting
Cov_1kernel_size =（3，3）	Batch size=64
Cov_1filters=16	$\alpha = 0.5$
Cov_2kernel_size=（1，1）	Learning rate=0.01
Cov_2filters=32	Hidden layer=1
Pooling Window Size=（2，2）	Hidden units=128
Activation Function=Relu	

The metrics selected to measure the prediction accuracy of the traffic states include the Mean Square Error (MSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE). The smaller these three error metrics, the higher the prediction accuracy. The calculation formulas for MSE, MAE, and MAPE are as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(v_i - \hat{v}_i)^2 \tag{24}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|v_i - v_i| \tag{25}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{v_i - \hat{v}_i}{v_i}\right| \times 100\% \tag{26}$$

where v_i represent the true value of the i-*th* velocity point, and $\hat{v}_\mathrm{i}$ represents predicted value of the i-*th* velocity point.

4.4. Parameter determination

To verify whether the Greenshield fundamental diagram model can accurately capture the intrinsic relationship between density and flow and further determine the fundamental diagram parameters. Traffic data at different sampling rates (10%, 20%, and 30%) on working days were selected for analysis in the experiment, as shown in Fig 6. It can be observed from the figure that both the overall data of the model and the data obtained under different sampling rates exhibit uneven distribution patterns, with most of the data concentrated in the density range of [0.05, 0.2]. This distribution is consistent with the density-flow relationship described in the Greenshield fundamental diagram. Therefore, the Greenshield fundamental diagram model can capture the relationship between density and flow, and the fundamental diagram parameters can be calculated accordingly.

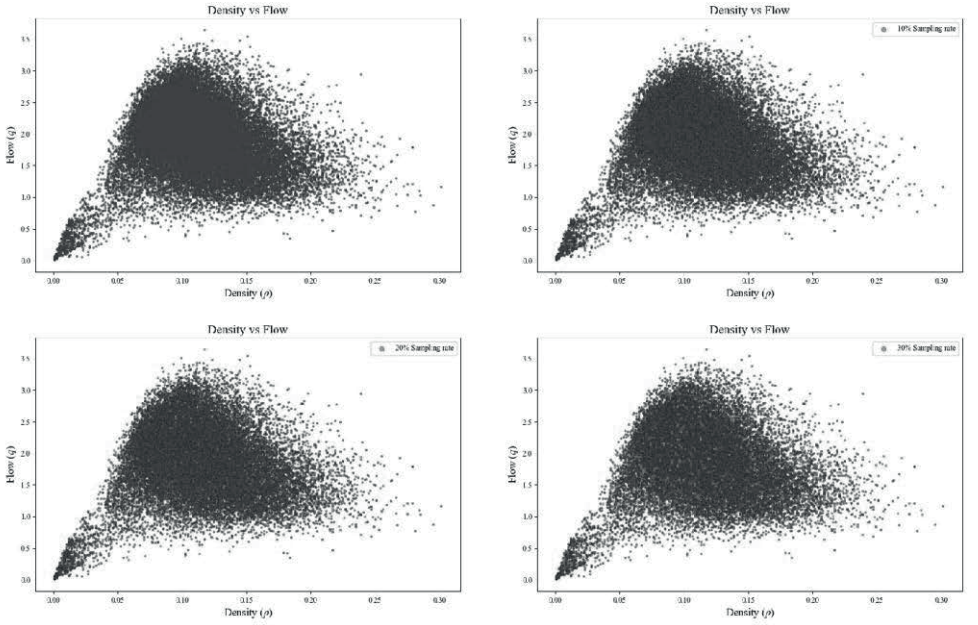


Fig. 6 - The density-flow values of the entire road section and the observed data under different sampling rates

Based on the method for determining the fundamental diagram parameters detailed in Section 3.4, the fundamental diagram parameters under the two scenarios (working and nonworking days) were calculated using different sampling rates (10%, 20%, and 30%). To evaluate the applicability and prediction accuracy of the fundamental diagram parameters across scenarios, traffic flow was predicted using the obtained parameters, and the MSE between the predicted and true flow values was calculated. The results are listed in Tables 4 and 5.

Tab. 4 - Fundamental diagram parameters on working days

Sampling rate	10%	20%	30%
v_f	51.45	51.12	46.64
ρ_j	0.17	0.17	0.18
MSE	19.63	19.69	19.82
R^2	0.912	0.915	0.909

Tab. 5 - Fundamental diagram parameters for non-working days

Sampling rate	10%	20%	30%
v_f	49.23	48.16	45.62
ρ_j	0.21	0.21	0.25
MSE	18.72	18.63	18.67
R^2	0.924	0.926	0.921

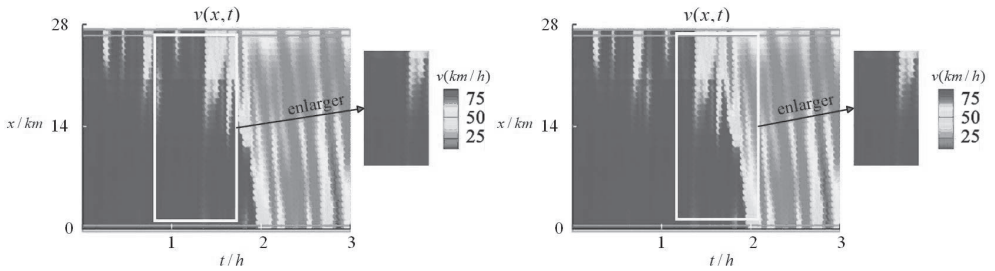
It can be seen from the table that MSE is less than 20 at all sampling rates, and the overall trend is relatively stable, which does not increase significantly with the increase of sampling rate. This indicates that the obtained fundamental graph parameters are somewhat robust and can adapt to data changes at different sampling rates. The R-squared is greater than 0.90 under different sampling rates, which fully proves the stability of our parameter estimation method.

4.5. Analysis of prediction results

4.5.1. Results analysis and performance evaluation of PI-CNN-LSTM model in traffic state predictions

(1) Analysis of the PI-CNN-LSTM prediction results

Following the predictions using real traffic data at a 10% sampling rate, a comparative analysis between the prediction results from the PI-CNN-LSTM method and the ground-truth traffic states is shown in Fig 7. Specifically, Figure (a) shows the real traffic states, whereas Figure (b) shows the traffic states predicted by the model, where different color gradients represent differentiated changes in the road vehicle velocity. In the yellow-boxed region of Figure (a), a clear color transition process is observed, which corresponds to the downward trend of vehicle velocity on actual roads and serves as an intuitive indicator of traffic congestion formation. Conversely, the same spatial location in Figure (b) exhibits a highly similar transition pattern, demonstrating that the PI-CNN-LSTM model can effectively capture the spatiotemporal characteristics of vehicle velocity variations in traffic flow and accurately predict the velocity attenuation process during congestion events. Further spatiotemporal dynamic analysis of the locally magnified region revealed that an abrupt velocity change occurring at a specific temporal node triggered the downstream diffusion of traffic congestion along the road. This complex traffic flow evolution feature is precisely mirrored in the prediction results on the right, which not only preserves the temporal node of the velocity mutation but also reasonably characterizes the spatial extent and propagation direction of congestion diffusion. The results confirm that the proposed PI-CNN-LSTM model can capture the dynamic evolution laws of the traffic state from a global perspective and enable fine-grained modeling of traffic flow phenomena with significant nonlinear characteristics, such as vehicle velocity decay and congestion propagation.

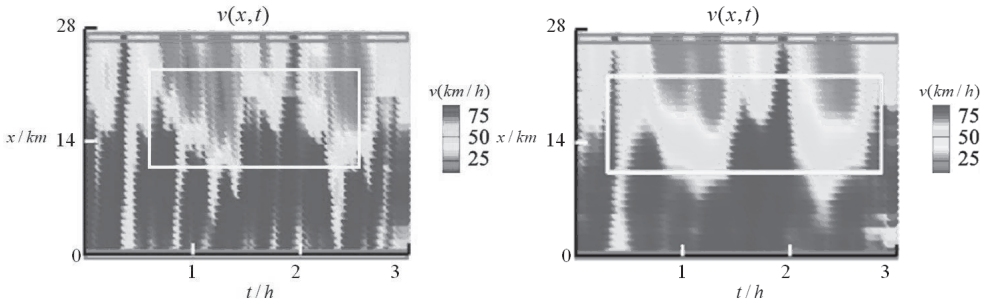


(a) True velocity (b) PI-CNN-LSTM prediction result

Fig. 7 - Results of PI-CNN-LSTM working days prediction

The PI-CNN-LSTM technique was tested in non-working-day scenarios to verify the model's universality across traffic circumstances. Fig. 8 shows the prediction results. Figure (a) shows ground-truth traffic states, while Figure (b) shows model-predicted traffic states with color

gradients representing road vehicle velocity variations. Traffic flow is very volatile on non-working days due to an extended spatial distribution of congested locations and increased formation frequency. In the yellow-highlighted section of Figure (a), a sharp velocity drop causes a color change, indicating local capacity deterioration due to unequal traffic demand distribution on non-working days. Figure b shows consistent velocity change trends at the same spatiotemporal regions, and the model accurately captures the beginning node of velocity attenuation and reasonably reflects congested areas' dynamic evolution. This shows that the PI-CNN-LSTM method, which incorporates the Greenberg fundamental diagram and LWR model, improves congestion area change capture, model performance, and adaptability under unstable traffic state conditions.



(a) True velocity (b) PI-CNN-LSTM prediction result

Fig. 8 - Results of PI-CNN-LSTM working days prediction

(2) Analysis of the results for different sampling rates

For full-segment traffic state prediction, the PI-CNN–LSTM technique incorporating traffic flow models was used after defining fundamental diagram parameters. The robustness of the model under sparse data settings was rigorously evaluated using observational data with three sample rates (10%, 20%, and 30%). Table 6 shows the experimental results. In the 10% low-sampling rate situation, the model's MSE was 13.24, MAE was 7.21, and MAPE was 6.98. When sample rate raised to 30%, MSE reduced by 1.19 to 12.05, a minor decrease. It appears that physical constraint mechanisms efficiently address data sparsity issues. By combining Greenberg fundamental diagram knowledge, the model produced stable predictions at 10% sampling rate. The table demonstrates that observation data availability grew with data sampling rate, demonstrating how observation data amount affects prediction results.

Tab. 6 - Errors at different sampling rates in working days scenarios

Sampling Rate	10%	20%	30%
MSE	13.24	12.89	12.05
MAE	7.21	6.56	6.19
MAPE	6.98	5.97	5.56

In the validation using the non-working-day datasets (Table 7), the PI-CNN-LSTM model demonstrated better adaptability to sparse data. At a 10% sampling rate, the MSE was 10.75, MAE was 2.27, and MAPE was 2.57, representing reductions of 2.49, 4.94, and 4.41%, respectively, compared with the errors at the same sampling rate in working-day scenarios. This performance