

Research Article

GRU-CNN Neural Network Method for Regional Traffic Congestion Prediction Serving Traffic Diversion Demand

Xue Xing , Xiaoyu Li , and Yaqi Zhai 

College of Information and Control Engineering, Jilin Institute of Chemical Technology, Jilin, China

Correspondence should be addressed to Xue Xing; xingx@jlicet.edu.cn

Received 4 July 2022; Revised 17 August 2022; Accepted 24 August 2022; Published 21 September 2022

Academic Editor: Kuruva Lakshmana

Copyright © 2022 Xue Xing et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Predicting spatiotemporal congestion situations of a traffic network is a prerequisite for urban traffic control. This study proposes a spatiotemporal traffic congestion situation prediction method based on the recurrent gated unit-convolutional neural network (GRU-CNN). Considering the time and space attributes of traffic data, the third-order tensor of the traffic data is extracted from the time domain, and the GRU is used to predict the traffic flow parameters of the traffic network. Then, the third-order tensor of multisource spatiotemporal traffic data is compressed into traffic data images and combined with the spatial structure. The feature extraction technology of a CNN is used to extract and identify the traffic network congestion features. Actual urban traffic network data are selected for model verification. The multistep prediction of the traffic flow parameters effectively ensures prediction accuracy. The proposed model is trained by the actual classification dataset. The prediction results of the test set demonstrate the model's reliability. Based on predicting the traffic parameters of the network, this model can give a highly accurate judgment of the traffic situation for the entire network. Compared with other models, the proposed model further improves the accuracy of road network traffic state discrimination and has better robustness.

1. Introduction

Although the traffic flow of an urban road network changes over a certain period, its basic structure is fixed for a long period [1]. Therefore, the changes in the traffic flow in the urban road network are closely related to the choices of travelers. Identifying traffic congestion conditions can help travelers and decision-makers quantitatively grasp the changes in traffic conditions and decide on path selection, road planning, and traffic control according to the changes [2, 3]. For the development of intelligent transportation, it is necessary to expand the analysis of traffic data characteristics for a single road section to the overall traffic state characteristics at the road network level [4].

This study takes an urban traffic network as a whole and analyzes its operation. The topology of the traffic network is abstracted based on the location and correlation of the detector. As the road network structure is considered fixed over a long period of time, the defining characteristic of urban traffic network change is the time-varying traffic flow, and the specific storage object is the traffic data, which are

continuously generated [5]. Due to the fixed location of the detection equipment, the acquired traffic data can match with the corresponding spatial information, and the changes in that data can depict the dynamic characteristics of the urban traffic network [6]. Therefore, considering the actual situation of the current large amount of traffic data, this study combines the constructed urban traffic network model and the description form of the traffic data tensor to determine the spatiotemporal state matrix of the dynamic road network and combines the time and space perspectives to predict traffic congestion. Considering the temporal and spatial attributes of the traffic data, third-order tensors of the traffic data are extracted from the time domain, and a GRU is used to predict the traffic flow parameters of the traffic network. Afterward, the third-order tensors of multisource spatiotemporal traffic data are compressed into traffic data images and combined with the spatial structure. A CNN is used to extract and identify traffic network congestion features. This paper combines the advantages of GRU and CNN to predict traffic parameters in the time domain and identify traffic states in the space domain and

finally provide information support for traffic diversion after regional traffic congestion identification.

2. Evaluation of Traffic Congestion in an Urban Road Network

A traffic congestion status is generally a category of traffic congestion within a selected area. It is used either to reflect the subjective feelings of people stuck in traffic or to meet the requirements for decision-making of management agencies. Specific grades are divided and described by the corresponding segments of common indicator variables or traffic parameters. The Road Capacity Manual published by the Transportation Research Board divides the traffic service level into six grades according to selected parameters. Germany proposed dividing the traffic status of an expressway into five levels according to the intervals of the traffic density [7]. In 2016, the Beijing Transportation Development Research Center proposed the Evaluation Index System of Urban Road Traffic Congestion, which classified different types of traffic status based on sections and defined road network congestion [8]. In this system, the traffic classification of sections is mainly based on the average speed of the section. The congestion degree of the road network is divided into very smooth, smooth, slightly congested, moderately congested, and severely congested.

There are two problems with the current standards of traffic congestion. First, there is only one single measurement index, and the state prediction lacks comprehensiveness. Second, there is a limited number of standards for classifying the traffic status of a road network, and the daily traffic congestion index ignores the changes in rules of the road network on each statistical day [9, 10]. Therefore, the five-level division of road network congestion is chosen as the classification of traffic congestion levels for the indicator system. However, two aspects should be added to the corresponding indicators. The first is to increase the comprehensiveness of the state measurements. Three common traffic parameters are used as the basis of measurements to jointly analyze the congestion situation. The second aspect is the percentage of traffic congestion mileage every 15 minutes, as shown in Table 1. It facilitates the analysis of daily traffic congestion changes in the road traffic network.

The calculation of the percentage of road congestion mileage every 15 minutes is as follows:

- (1) Taking 15 minutes as the statistical interval, the operation level of each road section is judged jointly according to the three traffic parameters. Clustering is used for the classification of the above levels
- (2) The percentage of road section mileage running in a severely congested road network is calculated (road section mileage is included as the coefficient)
- (3) The percentage of road network congestion mileage in 15 minutes is calculated by weighting the vehicle mileage (select the recommended value of the proportion of vehicle mileage in the index system)

TABLE 1: Classification of congestion levels in the road network.

Level	Road network congestion level	Percentage of road traffic congestion in 15 minutes
1	Very smooth traffic	[0%, 20%]
2	Smooth traffic	(20%, 40%)
3	Slightly congested traffic	(40%, 60%)
4	Moderately congested	(60%, 80%)
5	Severely congested traffic	(80%, 100%)

3. Image Extraction Method of Traffic Data

3.1. Construction of the Urban Traffic Network Model considering the Detector Layout. Each collected traffic dataset contains temporal and spatial attributes. Since the time attribute is the first-order attribute and the space attribute is a second-order attribute, the actual traffic data are represented as a third-order tensor.

Urban traffic is a complex, open, self-adaptive system characterized by abrupt changes [11]. Based on graph theory, the abstract network topology method for traffic feature extraction is discussed. According to the basic definition in graph theory [12], a graph $G = (V, E)$ consists of a set of nodes $V = \{p_1, p_2, \dots, p_n\}$ and a set of edges $E = \{e_1, e_2, \dots, e\}$ connecting pairs of nodes in V . If the graph contains the set of edge weights $W = \{w_1, w_2, \dots, w_n\}$ corresponding to the elements in E , the graph is called a weighted graph and denoted as $G = (V, E, W)$.

In this paper, the traffic data are from fixed detectors. Generally, fixed traffic detectors are placed at different locations where the flow of a section changes. Common locations are the entrances of traffic flow at intersections and ramps. To facilitate the loading of traffic data, the traffic network topology constructed in this section is mainly composed of road connection points connected through sections with limited lengths; that is, the original abstract method is used to construct the urban traffic network. In actual traffic operations, the traffic flow changes at the entrances and exits of intersections and ramps, so the traffic flow varies on the street sections in the urban network. A fixed point where the traffic flow state changes obviously is regarded as the connection point between sections, which is abstracted as a node in the urban road network. The network model adopts the directed graph structure in the original abstract method, where some parameters, including the road grade, number of lanes, and length, are not considered when extracting the topology. The dynamics of the network are determined by the traffic flow on the roads, which is embodied in the dynamic traffic data of the traffic network.

The traffic data entity is described as $\{t, \text{linkID}, \text{parameter}_1, \text{parameter}_2, \text{parameter}_3\}$, where t represents the time at which the detection data are collected, parameter_i represents the i -th traffic parameter in the detector, and $i = 1, 2, 3$ (the traffic detector can obtain many traffic parameters; however, only three common traffic parameters are selected). Common traffic data entities are described as $\{t, \text{linkID}, v, q, o\}$, where the corresponding time of the data is set as T , the location-averaged speed data are represented

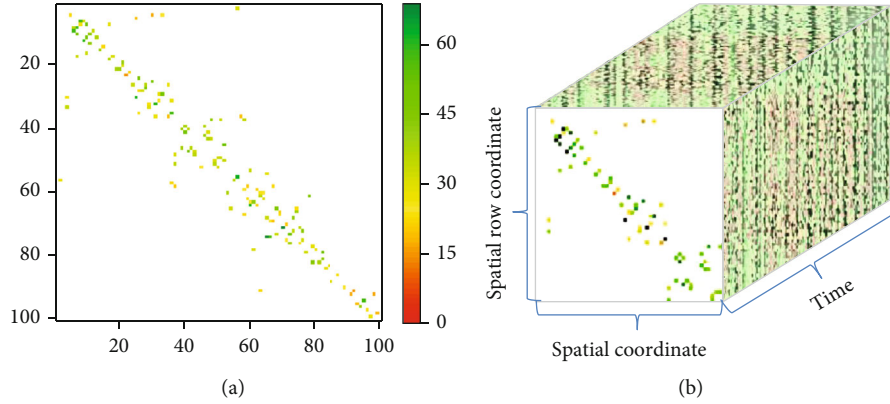


FIGURE 1: A graph of spatiotemporal traffic data tensors.

by v , the traffic flow data are represented by q , and the time-share data are represented by o . For the traffic network, V is the set of nodes, the included element $p_i (i \leq n)$ denotes the nodes, and p_i represents the location where the actual road traffic flow state changes significantly. E represents the set of road sections, the included element $e_j (j \leq k)$ denotes the edges, and e_j represents a one-way road section in the traffic network and corresponds to the road section in which the data collected by the detector $e_j = \langle p_{js}, p_{jo} \rangle$, where $p_{js} \neq p_{jo}$, and $p_{js}, p_{jo} \in V$. W is the set of edge weights in which $w_j (j \leq k)$ is the weight of edge e_j , and $w_j = w_{\langle p_{js}, p_{jo} \rangle}$ represents the traffic parameter data matched by the corresponding section $\langle p_{js}, p_{jo} \rangle$. To clarify the problem, three different parameters representing the traffic state are selected to form the traffic network $G = (V, E, W)$ with dynamically changing parameters. If $w_{\langle p_{js}, p_{jo} \rangle} = v_{\text{lim}} - v_{\langle p_{js}, p_{jo} \rangle}$ ($v_{\langle p_{js}, p_{jo} \rangle}$ is the location-averaged speed on section linkID corresponding to the sequence pair $\langle p_{js}, p_{jo} \rangle$ in the network within the same time segment, and v_{lim} represents the maximum speed limit on the section interval), the network can be expressed as $G = (V, E, W)$. If $w_{\langle p_{js}, p_{jo} \rangle} = q_{\langle p_{js}, p_{jo} \rangle}$ ($q_{\langle p_{js}, p_{jo} \rangle}$ is the traffic flow on section linkID corresponding to sequence pair $\langle p_{js}, p_{jo} \rangle$ in the network within the same time segment), the network can be expressed as $G = (V, E, W, q)$. If $w_{\langle p_{js}, p_{jo} \rangle} = o_{\langle p_{js}, p_{jo} \rangle}$ ($o_{\langle p_{js}, p_{jo} \rangle}$ is the time occupancy on section linkID corresponding to the sequence pair $\langle p_{js}, p_{jo} \rangle$ in the network during the same time segment), the network can be expressed as $G = (V, E, W, o)$.

3.2. Traffic Data Tensor Description Based on Traffic Network Structure. A tensor is a multidimensional array of data [13]. Tensors can be understood as extensions of vectors and matrices in a multidimensional space. Scalars, vectors, and matrices are representations of tensors in low-dimensional spaces: scalars are represented as zero-order tensors, vectors are first-order tensors, and matrices are second-order tensors. Each element in a tensor is associated with multiple indexes. The expression form of a tensor (A) is shown in Equation (1), where $n_i (i = 1, 2, \dots, m)$ is a positive integer and m is called the order of the tensor. If $n_1 =$

$n_2 = \dots = n_m = n$, then n is called the dimension of tensor A . An m -order tensor with n dimensions which are real numbers is denoted as $\Gamma(\mathbb{R}^n, m)$.

$$A = \left(a_{i_1, i_2, \dots, i_m} \right) \in \mathbb{R}^{n_1 \times n_2 \times \dots \times n_m}. \quad (1)$$

In order to clearly describe the tensor characteristics of the traffic data in the traffic network described above, spatial information of the traffic data is represented as a network adjacency matrix. The adjacency matrix is a matrix representing the network structure as in Equation (2), where $a_{p_{js}, p_{jo}}$ represents the element of the adjacency matrix M . In Equation (3), p_{js} and p_{jo} are node numbers of the traffic network, V , and the corresponding graphical diagram of M is shown in Figure 1(a). The third-order tensor of space-time traffic data is shown in Equation (3), where p_{js} and p_{jo} are node numbers of the traffic network, V , and t is the sequential number of traffic data collection time. The adjacency matrix corresponding to the urban traffic network weighted by the average road speed at a certain time is used to describe the network. Figure 1(a) is the heatmap of the adjacent matrix with 100 nodes in the network.

$$a_{p_{js}, p_{jo}} = \begin{cases} w_{p_{js}, p_{jo}}, & \langle p_{js}, p_{jo} \rangle \in E, \\ 0, & \langle p_{js}, p_{jo} \rangle \notin E. \end{cases} \quad (2)$$

$$A = \left(a_{p_{js}, p_{jo}, t} \right) \in \mathbb{R}^{n \times n \times T}. \quad (3)$$

The time attribute of the traffic data is combined with the spatial attribute of the road section, where the data is generated by being added to the adjacency matrix reflecting the weighted data. The data collected in the urban traffic network can be represented as a third-order tensor using three coordinates: the relevant intersections (nodes) of the data generated sections are taken as the spatial row and column coordinates, and the time when the data is generated is taken as the time coordinate. Figure 1(b) shows an example of the third-order tensor representing spatiotemporal traffic data.

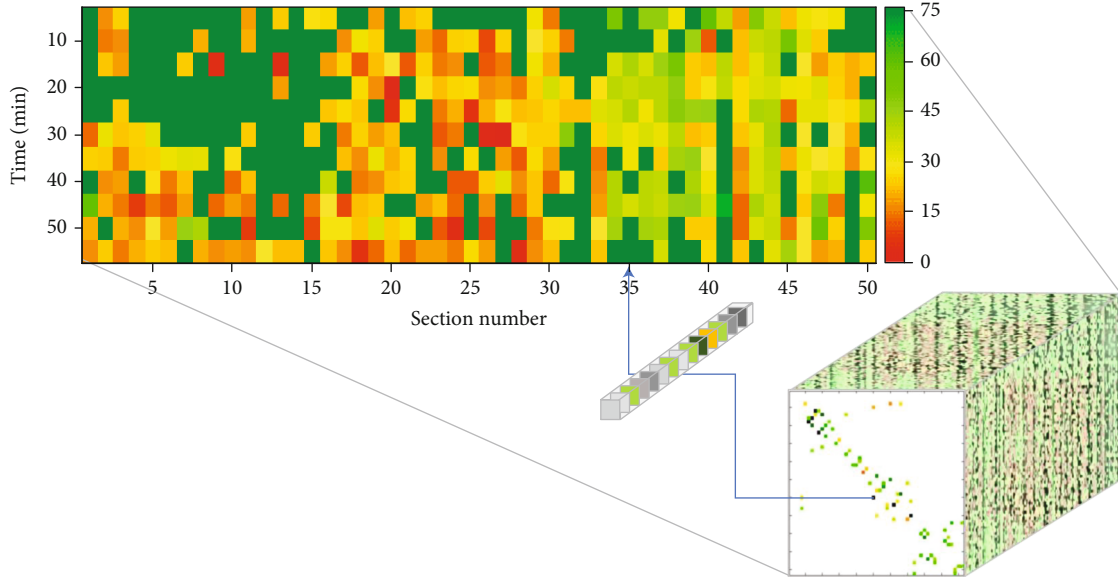


FIGURE 2: Spatiotemporal traffic data tensor compression extraction diagram.

3.3. Tensor Compression of Spatiotemporal Traffic Data Based on Tensor Tube Fibers. Extracting the tube fibers of the traffic data tensor maintains the temporal relationship in the traffic data. The corresponding tube fibers of each one-way road section are extracted from the third-order tensor of the space-time traffic data, and then, they are tiled and reconstructed to form a second-order tensor of the space-time traffic data. The row coordinates of each data recombination are labeled according to the sequence of the tensor tube fibers, and the column coordinates of each data recombination are labeled according to the sequence of the sections on the front of the tensor. The extraction process is illustrated in Figure 2. The front section of the third-order tensor of the spatiotemporal traffic single-parameter data is an image representation of the adjacency matrix corresponding to the traffic network. The elements of the adjacency matrix are shown in Equation (2). To extract the second-order tensor of the spatiotemporal traffic data, it is assumed that e represents the number of ordinal couples in E in the directed traffic network $G = (V, E, W)$ and s represents the total number of frontal slices of the third-order tensor of the spatiotemporal traffic data. The expression of elements $b_{j,k}$ in the second-order tensor $B = (b_{j,k}) \in \mathbb{R}^{e \times s}$ of the space-time traffic data is shown in Equation (4). The traffic data tensor is essentially a way of describing traffic data in both time and space dimensions. To extract the time characteristics of the traffic data, the data retaining the time sequence relationship are extracted first while ignoring the spatial relationship from the time dimension, and then, the extraction results are converted into the spatiotemporal traffic data tensor compression matrix.

$$b_{j,k} = \frac{255 \times w_{\langle p_{j_s}, p_{j_o} \rangle, k}}{w}. \quad (4)$$

In Equation (4), section j corresponds to the number of ordinal couples $\langle p_{j_s}, p_{j_o} \rangle$; that is, according to the order of

the edge in the adjacency matrix of the traffic network, the main order is the row, and the order is j . k represents the frontal slice number of the third-order tensor of the corresponding spatiotemporal traffic data, $w = \max(W)$. To facilitate image processing, $b_{i,k}$ is standardized, as shown in Figure 2.

3.4. Extraction of Traffic Data Tensor considering the Combination of Multiparameter Data. This section selects three parameters to describe the temporal and spatial traffic data together according to the actual detection situation and computational complexity. According to the extraction method described in the previous section, each parameter can be described as a second-order tensor B of the space-time relationship and $B = \{N_1, N_2, \dots, N_s\}$, where the row vector N_k is extracted at each time k . Considering the high performance of CNN in deep learning theory for image processing, a $\lceil \sqrt{e} \rceil \times \lceil \sqrt{e} \rceil$ matrix N' of traffic data based on time k is constructed, and the matrix data are provided by vector N_k . If the value of e of ordered pairs E in the original graph $G = (V, E, W)$ is less than the number of elements in the matrix, the corresponding zero-complement operation will be performed at the last position of the matrix.

The analysis of the traffic state mainly consists of the identification and evaluation of single-parameter data, such as the daily traffic congestion index. This data mainly determines the operation level of each section in the road network through the average travel speed at the section. Evaluation of single-parameter data is simple with low computational cost; however, the identification of the traffic state is not reliable. For example, if the actual state of the traffic flow on the road is free, the traffic state determined by the subjective driving speed of a driver is inconsistent with reality. Therefore, it is more reasonable to evaluate the traffic state by combining multiple commonly used traffic parameters. The image extracted from the time-based multiparameter traffic data is $C = (c_{i,j,h}) \in \mathbb{R}^{\lceil \sqrt{e} \rceil \times \lceil \sqrt{e} \rceil \times 3}$, where h represents the

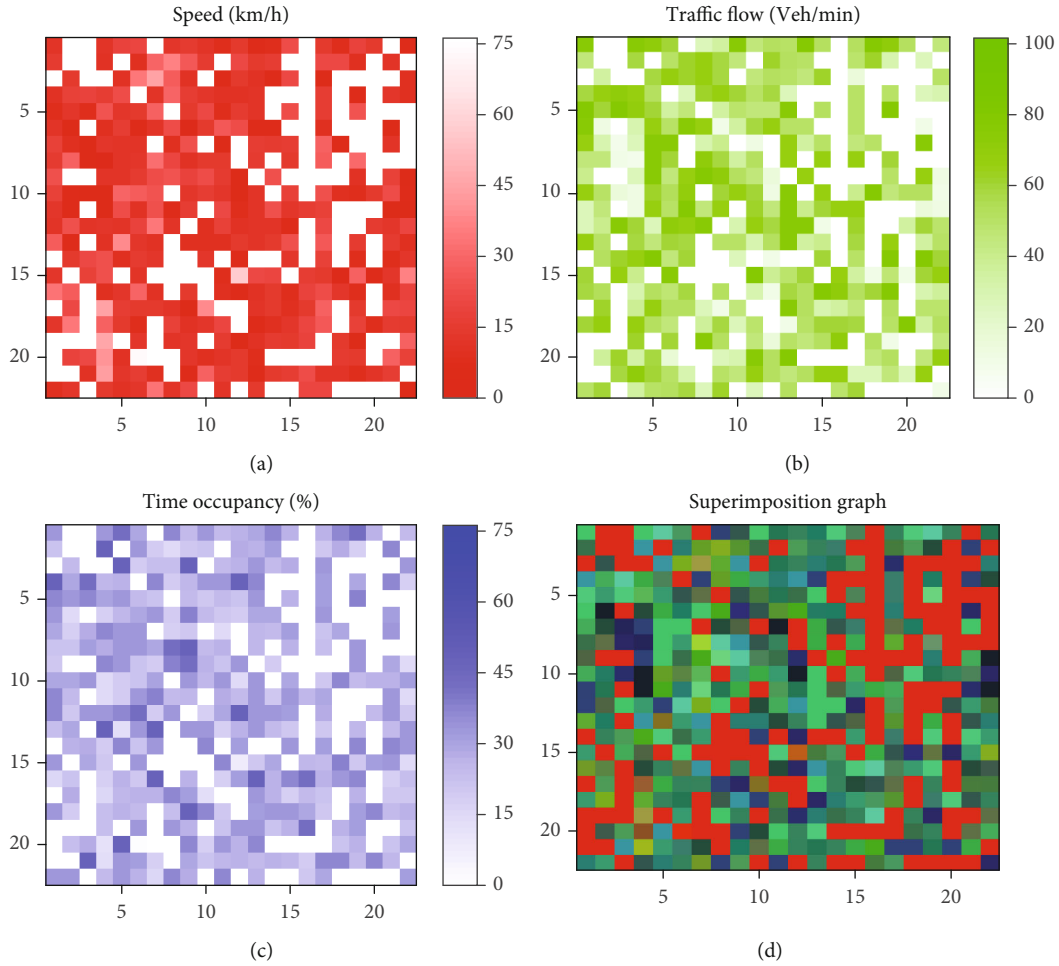


FIGURE 3: Third-order tensor diagram of spatial traffic data at a specific time point.

RGB color channel of the image: $h = 1$ represents the red channel (corresponding to the average traffic speed parameter $C_{::,1} = N'_{\nu}$), $h = 2$ represents the green channel (corresponding to the traffic flow parameter $C_{::,2} = N'_{\eta}$), and $h = 3$ represents the blue channel (corresponding to the traffic time occupancy parameter $C_{::,3} = N'_{\theta}$). An example of an RGB image is shown in Figure 3. The spatial traffic data in red, green, and blue are allocated according to the different traffic parameters, as shown in Figures 3(a)–3(c). The superposition of the three channels is shown in Figure 3(d). The “stacking” of multiple traffic parameters is realized by the three-layer channel stacking of the RGB images. This data representation is useful for identifying the network traffic state represented by multiple traffic parameters through image processing in subsequent steps.

4. Spatiotemporal Congestion Prediction for Road Networks Based on Deep Learning

A large amount of traffic data generated every day also plays a key role in improving the generalization ability of deep learning models [14]. With recent advances in architecture

design, deep learning has demonstrated the capabilities of fitting complex functions in various applications [15].

Deep learning is a subtype of machine learning developed based on neural networks. With a “deep” neural network structure, it overcomes multiple drawbacks of traditional machine learning techniques. The hierarchical structure of deep learning is composed of several layers between the input layer and the output layer, and a nonlinear information processing unit formed by these hierarchical structures can realize feature learning [16]. Considering this characteristic, deep learning extracts features from the original data through a deep neural network and eliminates the empirical setting of the original method. Many case studies have shown that high-level features extracted by deep neural networks are highly effective if sufficient training data exist.

Based on the prediction of traffic network parameters by a cyclic neural network, the congestion state of the spatial traffic network is identified to predict the overall situation of the traffic network. Figure 4 presents the temporal and spatial congestion situation and prediction model. The model consists of two parts: the time feature extraction of the traffic network using GRU and the congestion feature extraction of the traffic network using CNN.

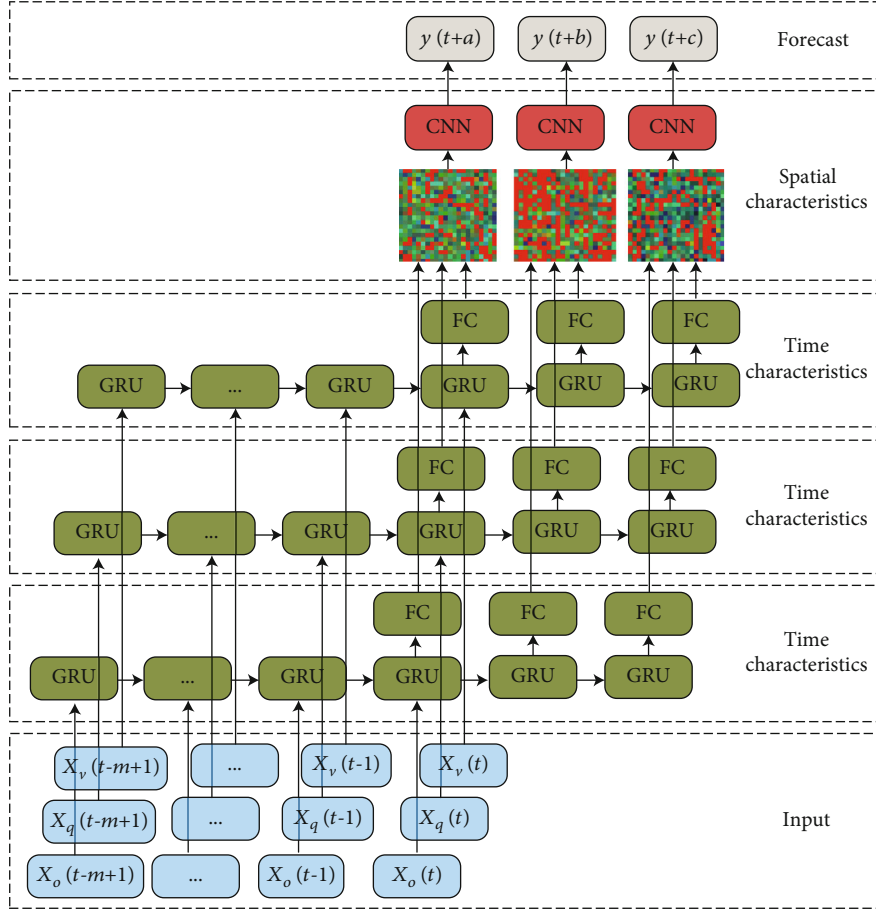


FIGURE 4: Frame diagram of space-time congestion prediction model.

5. GRU Model for Multistep Prediction of Traffic Flow

The first part of the prediction model uses the GRU of the neural networks to predict the network traffic parameters. Due to the congestion of the road network, more common evaluations of the road network traffic state parameter should be considered. Thus, a tensor is extracted from multiple parameters based on the road traffic data in time and space. And the corresponding parameters of the time series and the parallel GRU-assisted multistep iterative method are used to extract the time characteristics of the data for multistep prediction of the traffic parameters. The predicted traffic parameters will be used to identify the congestion state of the traffic network.

GRU is a variant of the long short-term memory (LSTM) neural network. It maintains the effectiveness of LSTM while having a simplified structure [17]. It merges the input gate and the forget gate in LSTM to form an update gate. GRU only contains an update gate and a reset gate. The update gate mainly determines the amount of hidden layer information from the previous time step that can be transferred directly to the current time step. The reset gate determines how much hidden layer information from the previous time step contributes to generating the current storage. Like

LSTM, GRU contains a gated unit that regulates the information flow inside the unit. GRU replaces the original self-updating storage state unit with a hidden state, which makes GRU more effective in data training.

Let the data be $\{x_t \in \mathbb{R}^n, y_t \in \mathbb{R}^m\}_{t=1}^T$, where x_t represents the input at time t and T is the length of the time series. The calculation of GRU is mainly to output the hidden state of each unit and the hidden state h_t^j of the j -th GRU at time t , as shown in Equation (5), where f is a linear transformation of the input vector $X = \{x_1, x_2, \dots, x_T\}$. The GRU algorithm contains four important parts.

$$h_t^j = f(h_{t-1}, x_t)^j. \quad (5)$$

Considering the basic time interval of the congestion evaluation and the time rule of the continuous expansion of traffic congestion, the time delay m of the model is determined. In addition, for the values of a , b , and c in the predicted congestion state, the three time steps a , b , and c correspond to the prediction results in this part; that is, the traffic parameter values of time steps a , b , and c after the current time step can be predicted using historical data.

Details of the algorithm are as follows:

- (Step 1) Identify the traffic situation according to three common parameters: the average speed, flow, and time occupancy.
- (Step 2) Extract the tube fibers (i.e., traffic parameter time series) of each among e sections from the third-order tensors corresponding to the three traffic parameters, determine the GRU model parameters $m, a, b,$ and $c,$ and group the time series of each section according to the model delay $m.$
- (Step 3) Extract the front section of each time interval Δt from the third-order tensors corresponding to the three types of traffic parameters.
- (1) Step 3.1: with a statistical interval of 15 minutes, judge the operation level of each section jointly according to the three traffic parameters.
 - (2) Step 3.2: set $k = 15/\Delta t,$ and calculate the mileage percentage of road sections running at grade 5 in the k frontal sections.
 - (3) Step 3.3: calculate the percentage of road network congestion mileage in 15 minutes by weighting the number of car kilometers.
 - (4) Step 3.4: according to the rules in Table 1, map out the corresponding road network traffic status of each group in the historical data.

(Step 4) Use the GRU traffic parameters to predict, and estimate a group of data test sets for each section of the road network.

- (1) Step 4.1: initialize the number of units and network structure, $j = 1,$ and the input vector $X = \{x_1, x_2, \dots, x_T\}$ in each group of data in the training set.
 - (i) Step 4.1.1: determine the input of the j -th GRU.
 - (ii) Step 4.1.2: calculate and execute the reset gate r_t^j in the j -th cell according to Equation (6), where W_r represents the input weight vector, U_r represents the cyclic weight vector of the reset gate, B_r represents the bias vector, x_t represents the input vector at time $t,$ and σ represents the sigmoid activation function ($\sigma(Z) = 1/(1 + e^{-Z})$).

$$r_t^j = \sigma(W_r x_t + U_r h_{t-1} + b_r)^j. \quad (6)$$

- (iii) Step 4.1.3: calculate and execute the candidate state \tilde{h}_t^j in the j -th unit according to Equation (7), where

\tanh represents the hyperbolic tangent function ($\tanh(z) = (e^{+z} - e^{-z})/(e^{+z} + e^{-z})$) and \odot represents vector multiplication.

$$\tilde{h}_t^j = \tanh(W x_t + U(r_t \odot h_t) + b_s)^j. \quad (7)$$

- (iv) Step 4.1.4: calculate and execute the update gate z_t^j in the j -th GRU according to Equation (8), where W_z represents the input weight vector, U_z represents the cyclic weight vector of the update gate, b_z represents the bias vector, x_t represents the input vector at time $t,$ and σ represents the sigmoid activation function.

$$z_t^j = \sigma(W_z x_t + U_z h_{t-1} + b_z)^j. \quad (8)$$

- (v) Step 4.1.5: obtain the output h_t^j of the j -th GRU using the following equation:

$$h_t^j = (1 - z_t^j) h_t^j + z_t^j \tilde{h}_t^j. \quad (9)$$

- (vi) Step 4.1.6: set $j = j + 1;$ if $j < =$ the number of units, repeat Step 4.1.2.
- (2) Step 4.2: establish the training model according to Step 4.1, and obtain the estimated value from the group execution in the test set.
- (3) Step 4.3: repeat Step 4.1 twice and obtain the corresponding estimate for each parameter.

Obviously, traffic data are time-dependent, and previous traffic flow states may still have a long-term influence on the current state. In order to have a prior grasp of the traffic congestion situation, it is necessary to determine the deployment of emergency plans some time ahead of the forecast time. To ensure the reservation time of decision and the prediction accuracy, a multistep iteration method is adopted, as shown in Equation (10), where m represents the embedding dimension of the data, the left side of each equation is the GRU input, the right side represents the predicted output value, and x_i represents the i -th row vector of matrix $B,$ $\{b_{1,i}, b_{2,i}, \dots, b_{e,i}\}.$ Because traffic parameters contain many types, $X_v, X_q,$ and X_o represent the average speed, traffic, and time occupancy data, respectively.

$$\begin{aligned}
X(t-m+1) &= (x_t, x_{t-1}, \dots, x_{t-m+1}) \longrightarrow \hat{x}_{t+1}, \\
X(t-m+2) &= (\hat{x}_{t+1}, x_t, \dots, x_{t-m+2}) \longrightarrow \hat{x}_{t+2}, \\
X(t-m+g) &= (\hat{x}_{t+g-1}, x_{t+g-2}, \dots, x_{t-m+h}) \longrightarrow \hat{x}_{t+g}.
\end{aligned} \tag{10}$$

$$\begin{cases} \Delta b_z = \sum_{t=1}^T \delta_{z,t}, \\ \Delta b_r = \sum_{t=1}^T \delta_{r,t}, \\ \Delta b_s = \sum_{t=1}^T \delta_{s,t}. \end{cases} \tag{15}$$

The forward propagation output in model training is shown in Equation (5). The error term of each gated unit is calculated using backpropagation. When $t = T$, the error of the GRU unit $j+1$ is reversely transmitted as δ_T^j , and when $t \in [0, T]$, δ_t^j can be expressed as in the following equation:

$$\delta_T^j = \begin{cases} \Delta \delta_t^{j+1}, & t = T, \\ \frac{\partial E}{\partial h_t^j} = \Delta \delta_t^{j+1} + \Delta \delta_{t+1}^j, & t \in [0, T], \end{cases} \tag{11}$$

where E is the sample loss, which can be obtained from the loss function at all times.

According to Equation (11) and the gating unit process, the error at each time-step can be computed as in Equation (12), where $\delta_{g,t} = \delta_t \odot z_t \odot (1 - \tilde{h}_t^2)$, $\delta_{t,t} = \delta_{s,t} \odot (U h_{t-1}) \odot r_t \odot (1 - r_t)$.

$$\delta_{t-1} = \Delta \delta_{t-1}^{j+1} + \delta_{z,t} U_z + (\delta_{s,t} \odot r_t) U + \delta_t \odot (1 - z_t). \tag{12}$$

Equations (13), (14), and (15) can be computed by accumulating the gradient at each time step, and the updated weight and bias gradient can be obtained.

$$\begin{cases} \Delta U_z = \sum_{t=1}^T \delta_{z,t} h_{t-1}, \\ \Delta U_r = \sum_{t=1}^T \delta_{r,t} h_{t-1}, \\ \Delta U = \sum_{t=1}^T \delta_{s,t} h_{t-1}, \end{cases} \tag{13}$$

$$\begin{cases} \Delta W_z = \sum_{t=1}^T \delta_{z,t} x_{t-1}, \\ \Delta W_r = \sum_{t=1}^T \delta_{r,t} x_{t-1}, \\ \Delta W = \sum_{t=1}^T \delta_{s,t} x_{t-1}, \end{cases} \tag{14}$$

The posterior transfer error term is calculated by the partial derivative of the loss function, as shown in

$$\delta_t^{j-1} = \frac{\partial E}{\partial h_t^{j-1}} = (\delta_{z,t} W_z + \delta_{r,t} W_r + \delta_{s,t} W) \odot f'(h_t^{j-1}). \tag{16}$$

6. CNN Model to Identify Regional Traffic Congestion

Since the traffic data tensor is compressed and extracted as an image, CNN should be a good choice as they are suitable for images. The principle of CNN is inspired by the human visual nervous system. Their basic structure includes an input layer, convolutional layer, pooling layer, fully connected layer, and output layer.

The second part mainly identifies the traffic states of the road network and refines the classification of the daily traffic congestion levels according to the original daily evaluation rules. Taking a time unit of 15 minutes as the investigation benchmark, the temporal and spatial traffic jam state characteristics of the whole road network are identified based on the predicted parameters to predict the temporal and spatial traffic jam situation. Multiparameter traffic data images are formed through the prediction data, and red, green, and blue spatial traffic data are allocated according to different traffic parameters. The three channels are superposed to form RGB images. It is beneficial to use the CNN method for images, particularly to predict the network traffic state represented by multiple traffic parameters. According to the classification of the traffic network congestion state corresponding to the 15-minute traffic congestion index in Section 2, the category which is identified by $y(t)$ is set to 5.

Since the traffic data tensor is compressed and extracted as an image, CNN should be a good choice as they are suitable for images. The principle of CNN is inspired by the human visual nervous system. Their basic structure includes an input layer, convolutional layer, pooling layer, fully connected layer, and output layer.

- (1) Convolution layer: the convolution operation mainly carries out feature abstraction of the input image and iterates some features extracted by every single layer through the multilayer superposition to obtain the complex features of the image. In the CNN, each convolution layer is composed of several convolution units, and the parameters of each convolution unit are optimized using the backpropagation algorithm. The basic formal expression of the convolution layer

is shown in Equation (17), where f is an activation function, l represents the current layer, M_j represents the corresponding convolution window checked at the j -th convolution, K represents the convolution kernel, and b is the bias of the current layer.

$$x_j^l = f \left(\sum_{i \in M_j} x_j^{l-1} K_{ij}^l + b_j^l \right). \quad (17)$$

- (2) Pooling layer: pooling operation performs the aggregation of space or feature types and reduces spatial dimension; that is, in a small area, a specific sample value is taken as the input value, as shown in Equation (18), where $\text{down}()$ is a sampling function, β represents the multiplier parameter of the current layer, and b is the bias of the current layer. Generally, features with large dimensions are obtained after the convolution layer. The features are split into several regions, and their maximum or average values (i.e., maximum pooling or average pooling, corresponding to the sampling function) are taken to obtain new features with small dimensions to reduce the parameters of the whole neural network.

$$x_i^j = f \left(\beta_j^l \text{down} \left(x_j^{l-1} \right) + b_j^l \right). \quad (18)$$

- (3) Fully connected layer: each fully connected node is connected with all the features of the previous layer, and all the extracted features are integrated. The full link layer based on the classification task is mainly responsible for training a classifier. The learned features are used as the inputs, and the output is the classification results.

The proposed CNN-based algorithm is as follows:

- (Step 1) Calculate and execute the estimated value of the road network prediction according to Equation (4), extract the road section form as $\lceil \text{sqrt}(e) \rceil \times \lceil \text{sqrt}(e) \rceil$ matrix N' , and superpose the matrix N' corresponding to the three parameters to obtain the traffic state image of the road network.
- (Step 2) Use the superimposed traffic state images obtained from the test set data grouping to identify the traffic state of the road network using CNN.
- (1) Step 2.1: initialize the weights of the network, determine the network hierarchy, and set the

number of the labels (identified categories) to 5 according to the traffic network congestion classification.

- (i) Step 2.1.1: according to the network hierarchy, pass the input data through the convolution layer, the pooling layer, the fully connected layer, and the softmax layer to obtain the output value.
- (ii) Step 2.1.2: calculate the error between the output value of the network and the target value.
- (iii) Step 2.1.3: when the error is greater than the expected value, transmit the error back to the network, and successively obtain the error of each layer; if the error is equal to or less than the expected value, go to Step 2.2.
- (iv) Step 2.1.4: update the weights according to the obtained errors and return to Step 2.1.2.
- (2) Step 2.2: obtain the training model according to Step 2.1, and execute the superimposed traffic state images derived from the test set data grouping to classify the road network traffic state.
- (Step 3) Output the road network traffic state classification; that is, obtain the road network traffic states through the forecast data.

7. Evaluation of the Algorithm

This section describes the implementation details and result analysis of the proposed method.

7.1. Instance Dataset. The traffic detection dataset of a medium-sized city in China is selected for model verification. The dataset contains the traffic detection data for each section from December 1 to December 31, 2014 (2763066 detection records). There are traffic detection records of 484 sections (one-way driving) in the road network. The overall topology of the road network is shown in Figure 5. First, the example dataset required by the experiment is extracted from the original traffic database using MySQL software. The dataset records, including the section ID, collection time, associated intersection, associated direction, average speed, traffic flow, and time occupancy, are obtained by screening. The instance dataset is divided into two parts: a training set and a test set. The dataset of the first 25 days of the instance dataset is used as the training set, and the dataset of the other six days is used as the test set.

Each type of traffic parameter in the data record is extracted from the actual data and stored in an independent data file. Each independent parameter dataset extracted is screened and eliminated by the verification method in traffic data preprocessing. Considering the influence of traffic lights

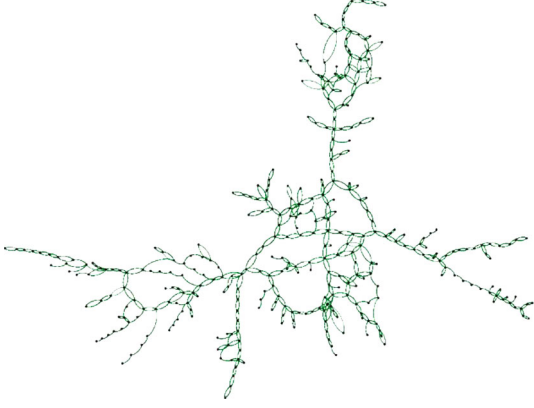


FIGURE 5: A structure diagram of the example city road network topology.

on urban road intersections, the time interval for fetching the fusion data when preprocessing the dataset is 5 minutes. The changes in the weighted adjacency matrix of the traffic flow parameters of the urban transportation network are shown in Figures 6–8. In the selected experimental dataset, the average traffic speed, traffic flow, and time occupancy rate changes at 50 intersections on December 1 were obtained from 6:00 to 7:30.

7.2. Experimental Programme Design. To evaluate the effectiveness of the temporal and spatial congestion prediction model based on the proposed GRU-CNN method, the model is verified according to the two structural parts in the model design.

The first part of the model predicts the parameters of each section of the traffic network, and the accuracy of the prediction determines the reliability of the subsequent traffic jam state prediction. To verify the accuracy of the prediction part in the model, the model is compared with common prediction methods, such as the autoregressive integrated moving average (ARIMA) method, the support vector machines (SVM), and the recurrent neural networks (RNN). For all the methods, considering the basic time interval of the congestion evaluation and the time law of the continuous expansion of traffic congestion, the changes in the traffic parameters in the first 30 minutes of the predicted time are used to predict the future values. In the evaluation of the algorithm, the average absolute percent error is selected to evaluate the prediction performance, mainly to compare the average absolute percent error of the predicted traffic flow parameter data for each group of data in the test set. The average absolute percent error is the average value of the absolute percent error, as shown in Equation (19), which can better reflect the actual proportion of the predicted difference.

$$\text{MAPE} = \frac{1}{m} \sum_{i=1}^m \left| \frac{y_i - \tilde{y}_i}{y_i} \right|, \quad (19)$$

where y_i represents the true value of the i -th road section corresponding to the test data group, \tilde{y}_i represents the

parameter predicted value of the i -th road section corresponding to the test data group, and m is the number of road sections in the experimental road network.

The second part of the model mainly identifies the overall state of the transportation network, extracts the images with multiparameter predicted values, and uses each group of the corresponding images as the input. To evaluate the classification accuracy, the model is compared with other common classification methods, such as the fuzzy clustering means (FCM), K -nearest neighbors (KNN), and support vector machines (SVM).

In addition to accuracy, precision, sensitivity, and specificity are also selected for model evaluation. The relevant definitions are shown in Equations (20)–(23). Among them, TP_i represents the number of samples that are classified as i ; TN_i is the number of samples classified as not i that are identified as the number of samples that are classified as not i ; FP_i represents the number of samples classified as not i that are identified as being classified as the number of samples of i ; FN_i is the number of samples classified as i that are recognized as samples of other classifications.

$$\text{Accuracy}_i = \frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i}, \quad (20)$$

$$\text{Precision}_i = \frac{TP_i}{TP_i + FP_i}, \quad (21)$$

$$\text{Sensitivity}_i = \frac{TP_i}{TP_i + FN_i}, \quad (22)$$

$$\text{Specificity}_i = \frac{TN_i}{TN_i + FP_i}. \quad (23)$$

7.3. Model Parameters. Python and Tensorflow are used to train and test the proposed framework. In the first part of the model, since the concentration time interval of the selected experimental data is 5 minutes, the time lag is set to $m=6$ based on the extraction of the historical data. According to the model setting $(a, b, c) = (1, 2, 3)$, since the data time interval is 5 minutes, the traffic parameters are predicted 5, 10, and 15 minutes after the input time. The training set and test set are grouped into one group every 30 minutes from the daily data; i.e., 283 groups of data are extracted for the daily traffic parameters of the network structure.

In the second part of the model, according to the actual road network extraction data, the first layer input in the convolution process is determined from the traffic data tensor (RGB image) of the multiparameter data combination. The settings of the convolution process are shown in Table 2.

The training process of the model relies on a forward propagation and a backpropagation step to achieve optimization. The forward propagation transmits hierarchical feature information, and the backpropagation updates the weights and bias vectors. The forward propagation computes the output of the GRU neural network (Equation (6)). The backpropagation of the GRU neural network uses the partial derivative of the loss function to calculate the backward

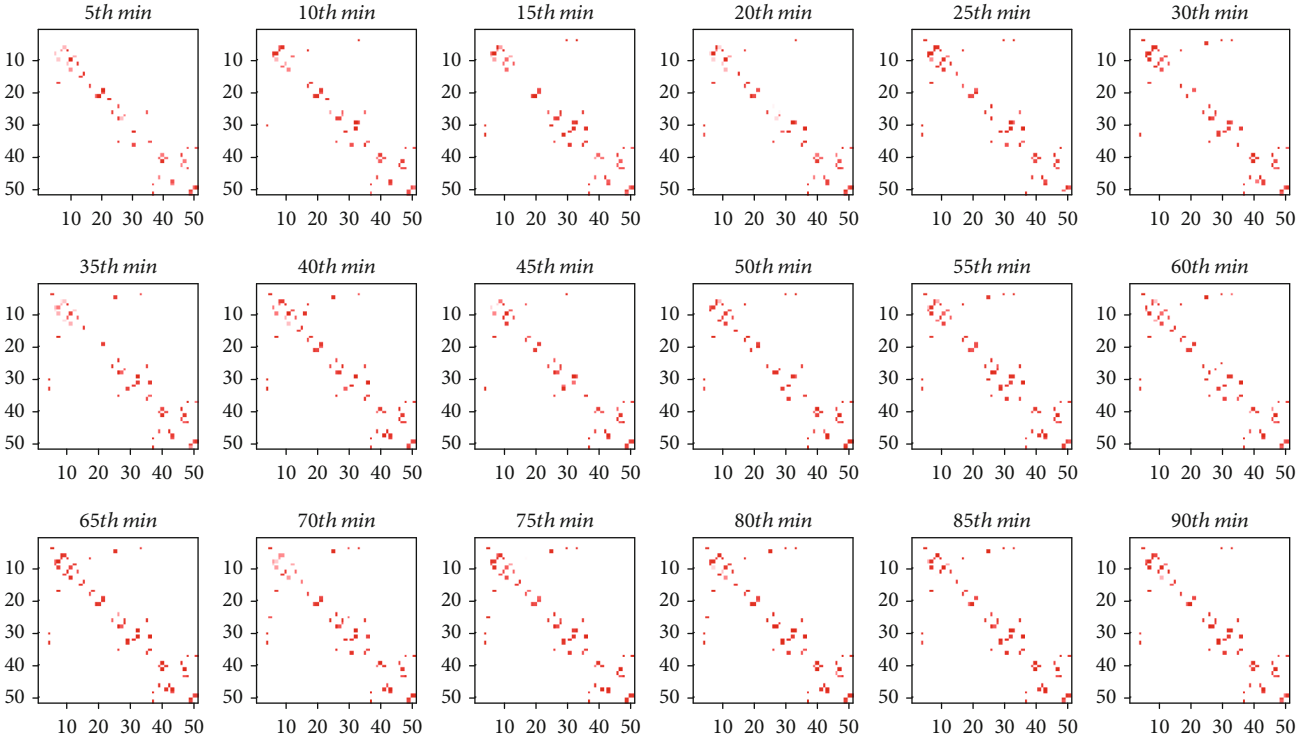


FIGURE 6: Weighted adjacency matrix diagram of road section speed in the urban transportation network.

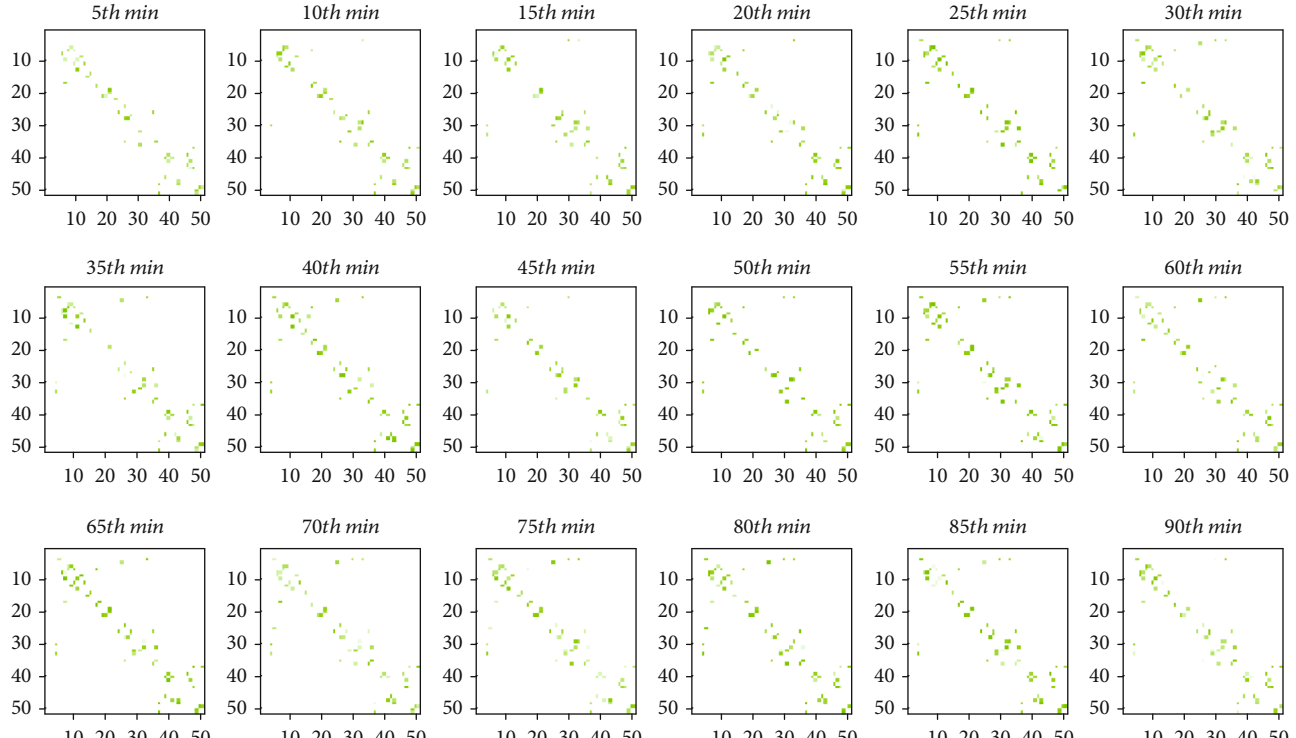


FIGURE 7: Weighted adjacency matrix diagram of traffic flow of road sections in the urban transportation network.

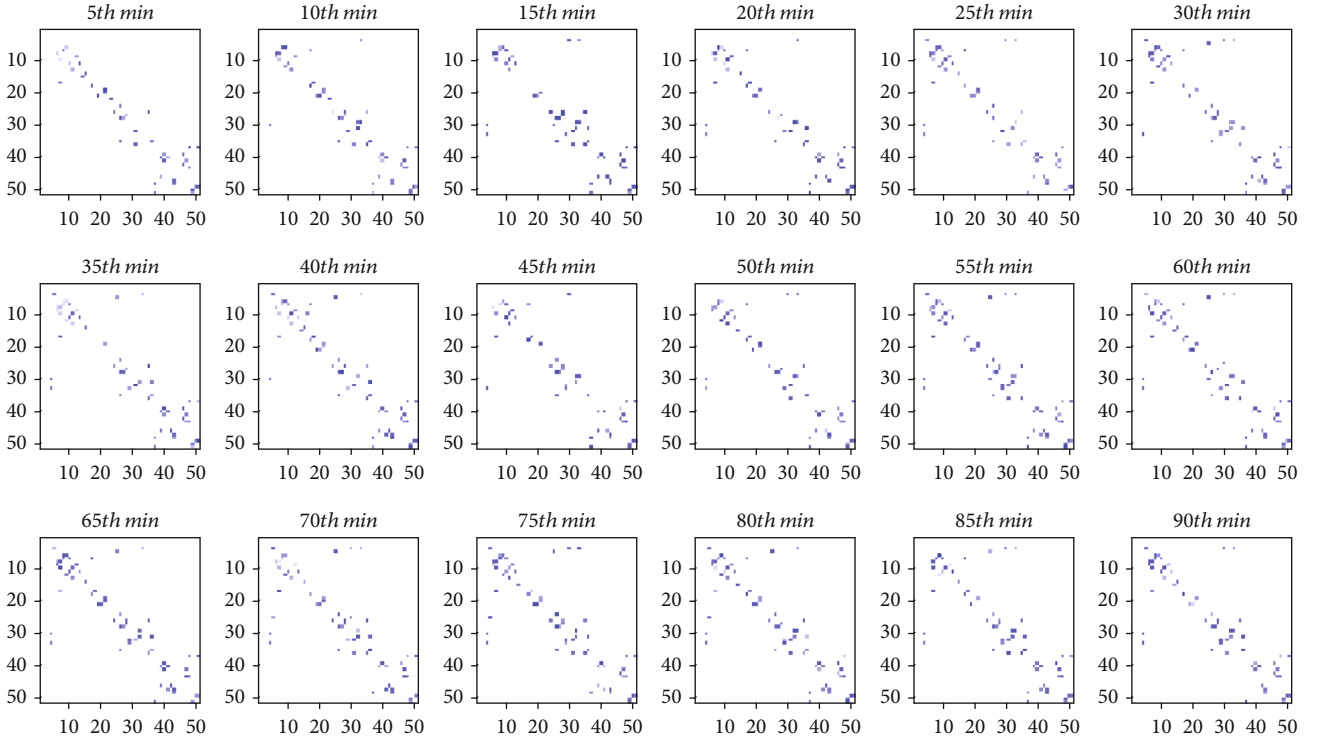


FIGURE 8: Weighted adjacency matrix diagram of time occupancy of road segments in the urban transportation network.

TABLE 2: CNN model structure based on multiparameter traffic data images.

Layers	Layer name	Channel	Filter size	Output
0	Input layer	3		22×22
1	Convolutional layer	16	$3 \times 3 \times 3$	$20 \times 20 \times 16$
		16	2×2	$10 \times 10 \times 16$
2	Maximum pooling layer	32	3×3	$8 \times 8 \times 32$
		—	—	—
3	Incentive (ReLU)	64	3×3	$6 \times 6 \times 64$
		64	2×2	$3 \times 3 \times 64$
4	Convolutional layer	—	—	576×1
5	Incentive (ReLU)	—	—	192×1
6	Convolutional layer	—	—	60×1
7	Maximum pooling layer	—	—	5×1

passing error term and the weight gradients and uses the weight gradients to update the weights using the gradient descent method. The forward propagation of the convolutional layer in the CNN model outputs $x_j^l = \text{ReLU}(\sum x_j^{l-1} K_{ij}^l + b_j^l)$. There is no activation function in the pooling layer. In forward propagation, this model selects the maximum pooling to compress the input. In the forward propagation of the output layer, the softmax activation function is used to calculate the classification probability. The backpropagation of the CNN is the same as above. The loss function measures the deviation between the network predictions

and the actual values. The smaller the loss function is, the more robust the model is. In the proposed method, the cross-entropy cost function is used as the loss function, and the gradient descent method is used to modify the optimized parameter values for the loss function so that the loss function tends to lower the error during the training process.

7.4. Experimental Results. The spatiotemporal traffic data tensors based on tube fibers are used to calculate the data of the training and test sets. Figure 9 shows the space traffic data tensor image for all the sections in the urban traffic

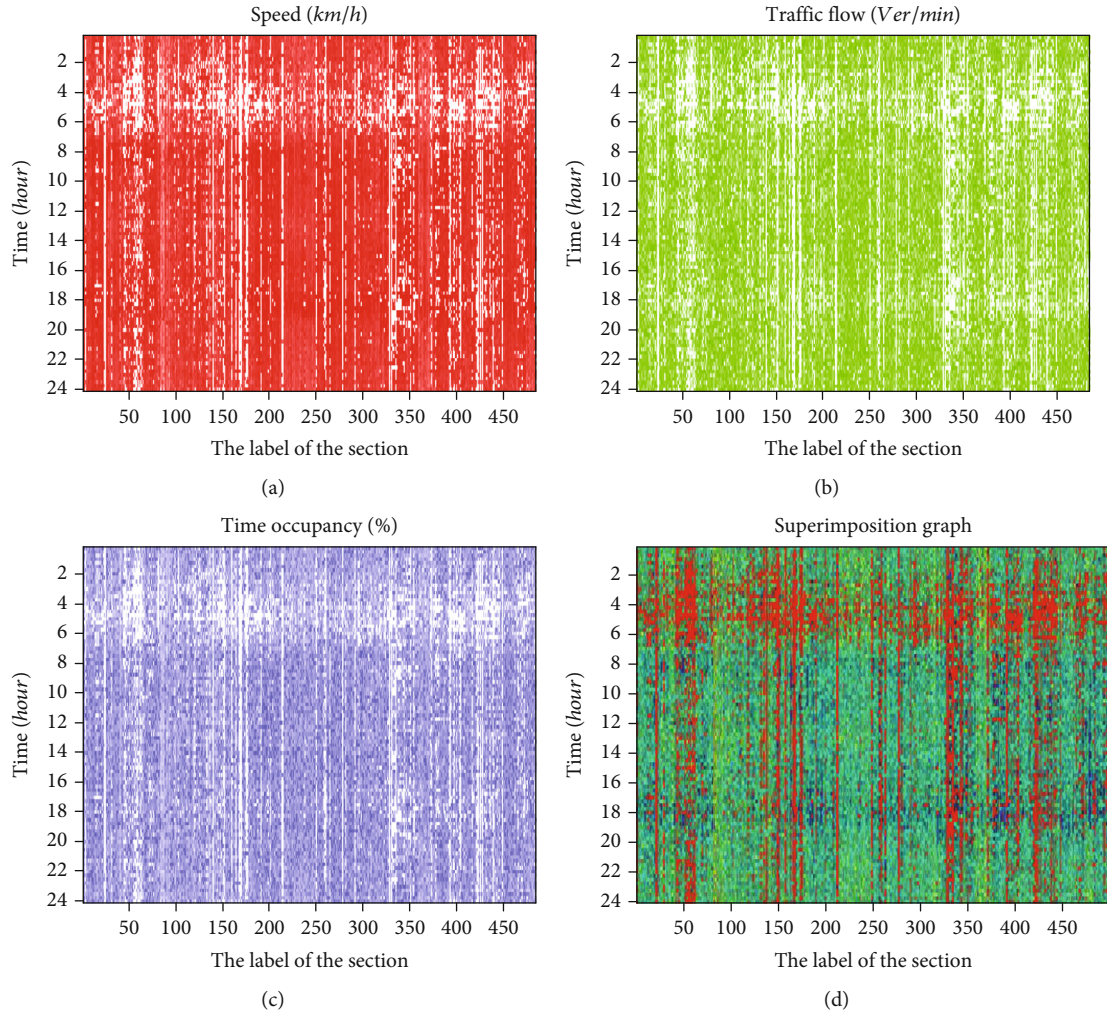


FIGURE 9: Spatiotemporal traffic data tensor images extracted from instance data.

network on December 1 for a duration of 24 hours. Figure 9(a) shows the average velocity data corresponding to the red channel of the image. Figure 9(b) shows the traffic flow data corresponding to the green channel of the image. Figure 9(c) shows the shared data corresponding to the blue channel image transportation time. Figure 9(d) shows the spatiotemporal traffic data tensor image of the three channels.

According to the scheme design in this section, the data of 484 one-way traffic sections in the experimental dataset are first verified for the prediction model of the traffic flow parameters. The prediction effects of the proposed model are compared with the ARIMA, SVM, RNN, and LSTM output for the same experimental data. According to the test set data for each type of traffic parameter, the corresponding traffic parameter prediction values of 5 min, 10 min, and 15 min are predicted, and the average absolute percent error of each group of data for each type of traffic parameter (including the average speed, traffic flow, and time occupancy according to the model design) is calculated.

The MAPE boxplot diagram of the experimental statistical data of each parameter is shown in Figure 10. The subfigures in the three rows in Figure 10 present the linear box

diagram of the average absolute percent error of the predictions on the test set data for each section for the average speed, traffic flow, and time occupancy, respectively. In the evaluation of each traffic parameter, the predicted and evaluated MAPE values are counted according to the prediction results at 5 min, 10 min, and 15 min. Figures 10(a), 10(d), and 10(g) correspond to the linear box diagram of the average absolute percent error of the 5 min prediction results. Figures 10(b), 10(e), and 10(h) correspond to the linear box diagram of the average absolute percent error of the prediction result at 10 minutes. Figures 10(c), 10(f) and 10(i) correspond to the linear box diagram of the average absolute percent error of the prediction result at 15 minutes.

It can be found from the comparison of the statistical graphs that the experimental overall error fluctuation of the cyclic neural network-based method is less than that of the ARIMA and SVM methods, and the overall experimental error of the 5 min prediction is lower than that of the longer prediction times. The following is a detailed analysis of the effects of the cyclic neural network models. The experimental data of the RNN, LSTM, and GRU methods are compared. The MAPE data of the 5 min prediction show that there is no significant difference among the three types of

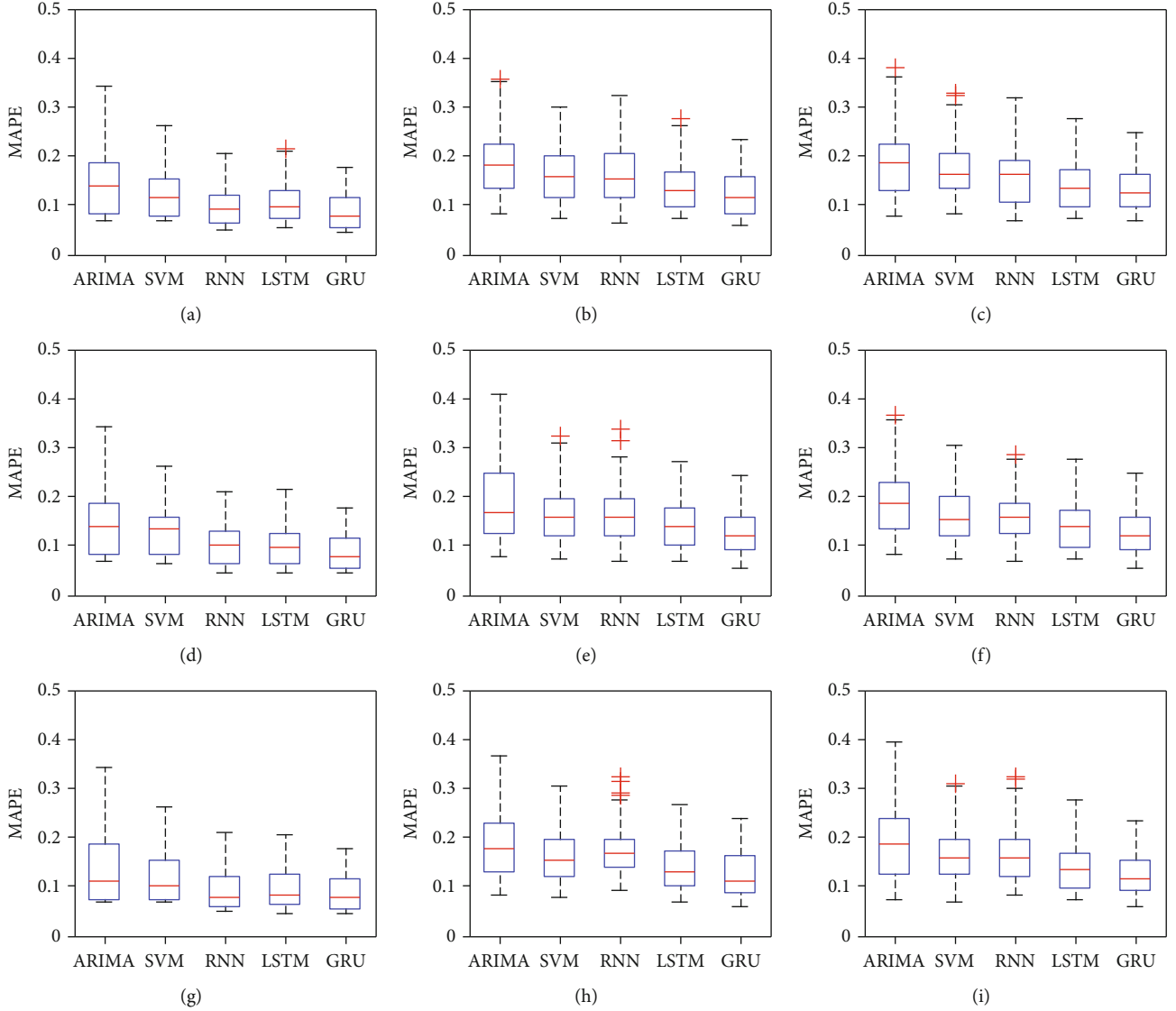


FIGURE 10: MAPE boxplot diagram of the experimental data.

traffic parameter data of the cyclic neural network models when they are predicted using the three methods. In fact, most of the methods in the cyclic neural network class are applicable. Compared with the MAPE data at 10 min and 15 min, the cyclic neural network method has better prediction stability than the RNN method when using the three methods to predict the three traffic parameters, which means both LSTM and GRU have advantages in prediction. From the perspective of the forecast duration and the category of the predicted traffic parameters, the method used in this model has the most stable fluctuation range of the average absolute percent error predicted by the actual traffic time series data, and thus, the proposed method is practical.

The CNN model is trained using the training set. It is then used on the test set to predict and extract the multiparameter traffic data image. The road network traffic state is predicted by the CNN image prediction algorithm. The images extracted from the predicted multiparameter traffic

data are shown in Figure 11. In that figure, 144 groups of $22 \times 22 \times 3$ images with the size of $22 \times 22 \times 3$ are composed of the superposition of the 15 min prediction parameters corresponding to 144×3 groups of data on December 29 in the test set.

The experiment is designed according to the hierarchical parameters of the model, and the images extracted from the multi-parameter traffic prediction data are used to predict the states of the traffic congestion types. The accuracy, precision, sensitivity, and specificity of each state prediction are evaluated. The second part evaluates the results of the first part of the model. This method is combined with the FCM, KNN, and SVM methods to predict the traffic status instantly. Using the FCM model in [18], the clustering centers of the flow [19 34 52 42 37] (Veh/min), speed [22 41 34 23 13] (km/h), and time occupancy [7 15 30 41 72] (%) are selected according to the experimental data. The membership degrees of the corresponding five types of road network

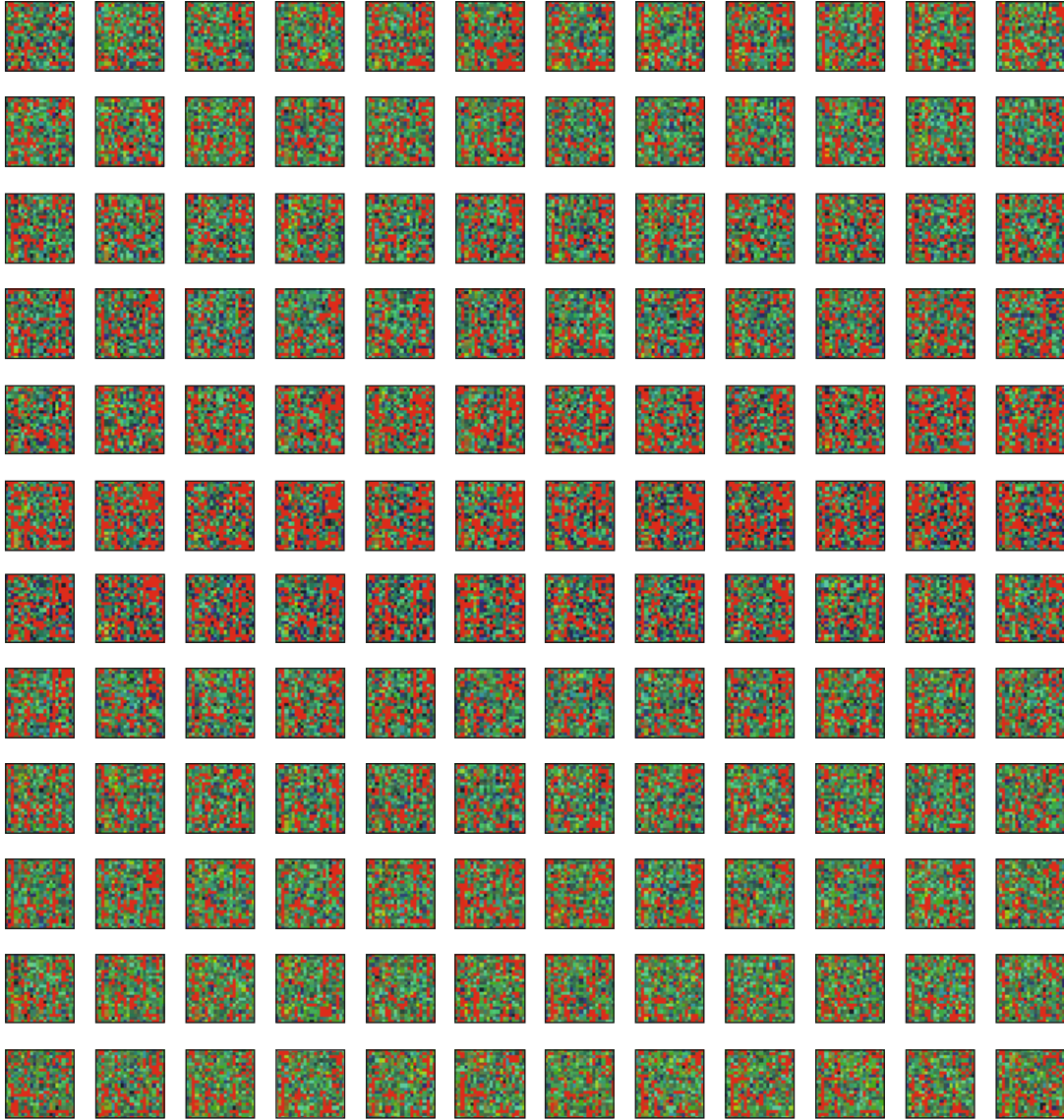


FIGURE 11: Image of multiparameter traffic data extraction based on the predicted values.

traffic states are [0.72 0.1 0.09 0.06 0.03], [0.24 0.6 0.1 0.04 0.02], [0.11 0.21 0.55 0.1 0.03], [0.02 0.08 0.1 0.65 0.15], and [0.02 0.14 0.09 0.19 0.56]. The KNN model in [19] is applied to set the number of adjacent points between 5 and 20. Based on the experimental data, the number of adjacent points is set to 15. The SVM model in [20] is used with the optimized RBF kernel function parameter of 2.4 and the penalty coefficient of 11.7. Figures 12(a)–12(d) present the confusion matrix corresponding to the test set for the CNN model, the FCM model, the KNN model, and the SVM model, respectively.

The confusion matrix reflects the classification performance of the model. The diagonal in the matrix is highlighted in cyan, where the values indicate the numbers of correct classifications and the corresponding percentages. The rows and columns corresponding to the other labels are the numbers and percentages of the row classifications assigned to the column classification, respectively. The dis-

criminant rate and misjudgment rate of each classification correspond to the grey cells in each row and column. The data in the lower right corner of the matrix are the overall accuracy and misjudgment rate of all state classifications of the model. Taking Figure 12(a) as an example, the numbers of the correctly classified traffic conditions of the five road networks are 730, 333, 257, 136, and 88, with the corresponding percentages of 43.0%, 19.6%, 15.1%, 8.0%, and 5.2%, respectively. In this section, the accuracy rate and misjudgment rate of the traffic status judgment of the whole road network using the model are 90.9% and 9.1%, respectively.

By comparing all the confusion matrices, the overall accuracy of the KNN and SVM models is 1.3%, and that of the FCM and SVM models is 0.3%, while the CNN model produces the highest overall accuracy. In the comparison of the misjudgment rate of road network traffic conditions, although the number of misjudgment types accounts for a

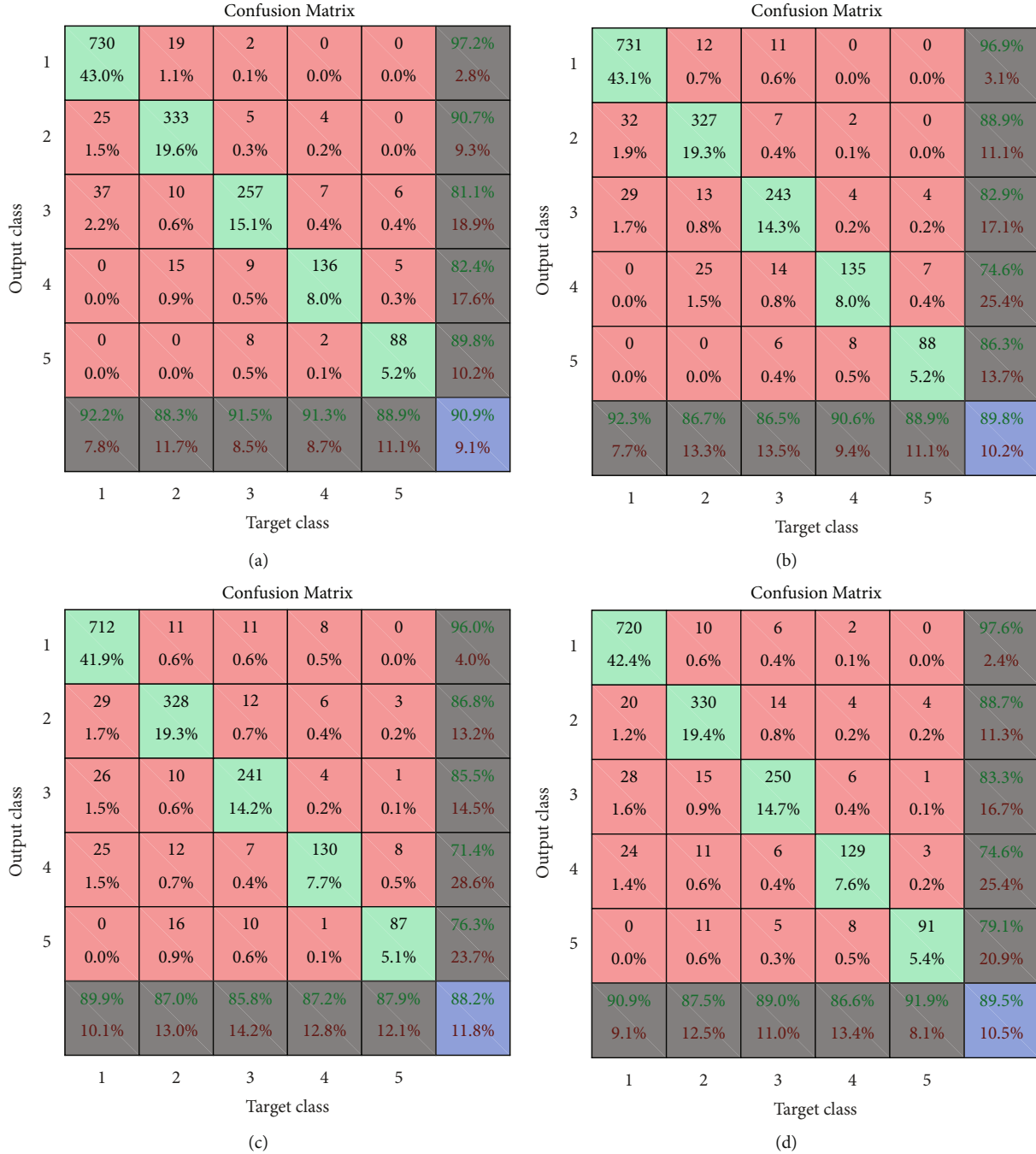


FIGURE 12: The confusion matrix of each model.

small proportion, the misjudgment types of the CNN and FCM models are close to the original categories. In contrast, the misjudgment types of the KNN and SVM models are more scattered, indicating serious category misjudgment. This shows that the model proposed herein achieves better stability in the identification of road network traffic states. Among the comparison results of the four models, in the classification of moderate congestion and severe congestion, the misjudgment rate obtained by each model is high, which is due to the relatively small proportion of the training set data in this category in the actual network classification.

8. Conclusions

The prediction results of the overall experimental parameters show that the spatiotemporal congestion situation prediction model based on GRU-CNN can predict the traffic parameters of the whole network with a highly accurate judgment of the traffic situation. Compared with other models, the accuracy of the road network traffic state (i.e., the road network traffic situation) of the proposed model is improved, and the proposed model is more robust. Due to the limitation of collecting data samples, the current

model is developed with insufficient training data. Its performance should be improved when more data are collected.

This manuscript is part of the supporting technology for the prediction of spatiotemporal traffic data in the analysis of urban traffic network dynamics. This study takes the whole urban traffic network as the analysis object and uses deep learning technology to extract the characteristics of the network to predict the congestion situation.

Using an urban traffic network model and traffic data tensors, this study determines the space-time state matrix of the dynamic road network and predicts the road network congestion situation from the perspectives of time and space. From the point of view of the time dimension, there is a strong correlation in the traffic time series, and the gated cycle unit neural network is used to predict the traffic network parameters using historical data. From the perspective of space, the locations where congestion often occurs are fixed, so the CNN is used to identify the congestion state of the spatial traffic network based on the predicted values of the overall situation. Effectively predicting the trend of traffic network operation is helpful for planning the deployment of traffic control and releasing early warnings and guidance information.

The study is mainly based on the specific road network data and traffic detection data of a medium-sized city, so the main results of the research are limited to the road network of medium-sized cities. There are a large number of complex interchanges in the road network structure in large cities. Since there is no data information corresponding to large-scale cities, the conclusions of this study cannot be extended to the traffic problems of large-scale cities. In the next stage, we will combine the previous research conclusions, aiming at the prediction of traffic congestion areas and fully considering the congestion avoidance mechanism when providing corresponding route recommendations to travelers. This is to alleviate the congestion level of the overall traffic network and improve the efficiency of the traffic flow in road network operation.

Data Availability

The traffic detection dataset of a medium-sized city in China is selected for model verification. The dataset is provided by the project sponsor, and it is confidential. The dataset contains the traffic detection data of each road section from December 1 to December 31, 2014 (2763066 detection records), with 484 sections (one-way driving) in the urban road network.

Conflicts of Interest

The authors declare no conflict of interest.

Acknowledgments

This research has been jointly supported by the Science and Technology Development Plan Project of Jilin Province (Natural Science Foundation of Jilin Province, Grant No. 20210101416JC).

References

- [1] S. Pan, P. Li, C. Yi, D. Zeng, Y. C. Liang, and G. Hu, "Edge intelligence empowered urban traffic monitoring: a network tomography perspective," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 4, pp. 2198–2211, 2021.
- [2] M. Mohan and S. Chandra, "Capacity estimation of unsignalized intersections under heterogeneous traffic conditions," *Canadian Journal of Civil Engineering*, vol. 47, no. 6, pp. 651–662, 2020.
- [3] V. Najafi Moghaddam Gilani, S. M. Hosseinian, M. Ghasedi, and M. Nikookar, "Data-driven urban traffic accident analysis and prediction using logit and machine learning-based pattern recognition models," *Mathematical Problems in Engineering*, vol. 2021, Article ID 9974219, 11 pages, 2021.
- [4] R. Zhou, H. Chen, H. Chen, E. Liu, and S. Jiang, "Research on traffic situation analysis for urban road network through spatiotemporal data mining: a case study of Xi'an, China," *IEEE Access*, vol. 9, pp. 75553–75567, 2021.
- [5] M. P. H. Raadsen, M. C. J. Bliemer, and M. Bliemer, "Steady-state link travel time methods: formulation, derivation, classification, and unification," *Transportation Research Part B: Methodological*, vol. 122, pp. 167–191, 2019.
- [6] M. Saeedmanesh and N. Geroliminis, "Dynamic clustering and propagation of congestion in heterogeneously congested urban traffic networks," *Transportation Research Part B: Methodological*, vol. 105, pp. 193–211, 2017.
- [7] Q. Bing, B. Gong, Z. Yang, C. Lin, and X. Qu, "Traffic state identification for urban expressway based on projection pursuit dynamic cluster model," *Journal of Southwest Jiaotong University*, vol. 50, pp. 1164–1169, 2015.
- [8] L. I. N. Ping, Y. U. Xun-yi, L. I. U. Fang, and L. E. I. Zhen-ming, "A network traffic classification algorithm based on flow statistical characteristics," *Journal of Beijing University of Posts and Telecommunications*, vol. 31, no. 2, pp. 15–19, 2008.
- [9] Z. Cheng, W. Wang, J. Lu, and X. Xing, "Classifying the traffic state of urban expressways: a machine-learning approach," *Transportation Research Part A: Policy and Practice*, vol. 137, pp. 411–428, 2020.
- [10] Z. Chen, Y. Jiang, and D. Sun, "Discrimination and prediction of traffic congestion states of urban road network based on spatio-temporal correlation," *IEEE Access*, vol. 8, pp. 3330–3342, 2020.
- [11] N. Geroliminis and I. Sun, "Properties of a well-defined macroscopic fundamental diagram for urban traffic," *Transportation Research Part B: Methodological*, vol. 45, no. 3, pp. 605–617, 2011.
- [12] J. A. Bondy and V. Chva'tal, "A method in graph theory," *Discrete Mathematics*, vol. 15, no. 2, pp. 111–135, 1976.
- [13] B. Cyganek, "Thumbnail tensor—a method for multidimensional data streams clustering with an efficient tensor subspace model in the scale-space," *Sensors*, vol. 19, no. 19, p. 4088, 2019.
- [14] A. Miglani and N. Kumar, "Deep learning models for traffic flow prediction in autonomous vehicles: a review, solutions, and challenges," *Vehicular Communications*, vol. 20, no. 12, p. 100184, 2019.
- [15] G. Montavon, W. Samek, and K. R. Müller, "Methods for interpreting and understanding deep neural networks," *Digital Signal Processing*, vol. 73, pp. 1–15, 2018.

- [16] K. Wongsuphasawat, D. Smilkov, J. Wexler et al., “Visualizing dataflow graphs of deep learning models in TensorFlow,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 1–12, 2018.
- [17] R. Dey and F. M. Salemt, “Gate-variants of Gated Recurrent Unit (GRU) neural networks,” in *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pp. 1597–1600, Boston, MA, USA, August 2017.
- [18] N. Lei, G. D. Zhang, and L. J. Chen, “Evaluation of traffic status classification for urban expressway based on improved fuzzy clustering,” *Highway*, vol. 62, pp. 134–139, 2017.
- [19] D. J. Nair, F. Gilles, S. Chand, N. Saxena, and V. Dixit, “Characterizing multicity urban traffic conditions using crowd-sourced data,” *PLoS One*, vol. 14, no. 3, p. e212845, 2019.
- [20] R. Yu, G. Wang, J. Zheng, and H. Wang, “Urban road traffic condition pattern recognition based on support vector machine,” *Journal of Transportation Systems Engineering and Information Technology*, vol. 13, no. 1, pp. 130–136, 2013.