

Research Article

How to Identify Patterns of Citywide Dynamic Traffic at a Low Cost? An In-Depth Neural Network Approach with Digital Maps

Li Zhang ¹, Ke Gong ¹, Maozeng Xu,¹ Aixing Li,² Yuanxiang Dong,³ and Yong Wang¹

¹School of Economics and Management, Chongqing Jiaotong University, Chongqing 400074, China

²School of Information Science and Engineering, Chongqing Jiaotong University, Chongqing 400074, China

³School of Management Science and Engineering, Shanxi University of Finance and Economics, Taiyuan, Shanxi 030006, China

Correspondence should be addressed to Ke Gong; gks_cn@163.com

Received 23 November 2020; Revised 25 April 2021; Accepted 6 May 2021; Published 24 May 2021

Academic Editor: Guang Li

Copyright © 2021 Li Zhang 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.

The identification and analysis of the spatiotemporal dynamic traffic patterns in citywide road networks constitute a crucial process for complex traffic management and control. However, city-scale and synchronal traffic data pose challenges for such kind of quantification, especially during peak hours. Traditional studies rely on data from road-based detectors or multiple communication systems, which are limited in not only access but also coverage. To avoid these limitations, we introduce real-time, traffic condition digital maps as our input. The digital maps keep spatiotemporal urban traffic information in nature and are open to access. Their pixel colors represent traffic conditions on corresponding road segments. We propose a stacked convolutional autoencoder-based method to extract a low-dimension feature vector for each input. We compute and analyze the distances between vectors. The statistical results show different traffic patterns during given periods. With the actual data of Chongqing city, we compare the feature extraction performance between our proposed method and histogram. The result shows our proposed method can extract spatiotemporal features better. For the same data set, there is little difference in the number distribution of red pixels found in the statistics result of the histogram, while differences do exist in the results of our proposed method. We find the most fluctuated morning is on Friday; the most fluctuated evening is on Tuesday; and the most stable evening is on Wednesday. The distance captured by our method can represent the evolution of different traffic conditions during the morning and evening peak hours. Our proposed method provides managers with assistance to sense the dynamics of citywide traffic conditions in quantity.

1. Introduction

Traffic condition prediction is a vital problem in an intelligent transportation system (ITS). Improving prediction performance in a large-scale urban network is difficult because of its highly dynamic and complex nature [1–3]. It is seen that insights into traffic patterns can improve the prediction accuracy of travel conditions [4]. Besides, tracking those changes in traffic patterns helps to assess the impacts of city construction planning and policy implementation [5], especially for peak hours. Once there are some traffic restriction policies, drivers may change their routes [6] or commuting mode [7] during peak hours. From citywide perspective, drivers' decisions will bring changes in the distribution of traffic flows in the network and thus

induce pattern fluctuation. Managers need to detect and measure such fluctuations to adjust some auxiliary traffic control decisions for those policies, such as deployment of traffic policies and route maintenance engineering. It has aroused a lot of attention in recent years to study traffic flow patterns within a scale of road networks, and there are still two research gaps to explore further.

First, the measurements of citywide traffic patterns are not limited to discrete and discontinuous flow of traffic. Traditionally, scholars define and study traffic patterns based on historical data of traffic flow metrics, such as speed [8], daily flow [9, 10], volume [11], density [12], and mobile trajectories [13]. In the beginning, these traffic flow data are collected using fixed-pointed detectors. Since the deployment of these sensors is of high cost and is geographically

discrete, traffic flow data from them cannot cover the whole city. Subsequently, communication data and social media data are employed as supplements. The data from floating cars, GPS, or social media only cover limited areas depending on the trajectories of moving objects, and they are usually not continuous because the communication has broken down. These raw traffic data may not be sufficient to explain the citywide pattern changes. However, digital maps integrate different types of data and generate real-time, city-scale traffic condition data continuously [14]. Digital maps use different colors to denote variable traffic conditions on corresponding roads. The data are public, real-time, temporal, and spatial. Thus, digital maps may provide ideal integrated data to analyze the citywide traffic pattern changes.

Second, studies start from local to road networks and utilize global traffic states, that is, they apply multiple dimension reduction techniques to aggregate the traffic states of all links in an entire urban road network and thus extract and analyze the specific traffic patterns. In images, there are kinds of methodologies to quantize features of them, such as histogram quantization [15], principal component analysis [16], and clustering [17–19]. In terms of pixel level, there shall be a high-dimensional sparse vector or matrix for a digital map. It is a challenge to capture its main feature. However, Boquet et al. [20] have proved that there is a low-dimensional latent space containing the underlying characteristics of the urban traffic. They propose a generative model to learn how traffic data are generated and inferred. They do not pay attention to the citywide pattern evolution. We define the evolution based on distance shifts of feature vectors for the series of maps. These feature vectors shall reflect the color spatial distribution characteristics of these maps. The convolutional neural network (CNN) is commonly used to extract images' spatial features [21], including colors [22]. Autoencoder has presented better insights into representation learning. Based on digital maps, we proceed with a stacked convolutional autoencoder to extract feature vectors.

To address the above research gaps, we propose a framework to identify the spatiotemporal evolution of citywide traffic through the proven latent space. The main contributions are twofold.

- (1) We introduce integrated, real-time, city-scale digital maps as the input. The digital maps are open and cheaply accessible and cover the entire city. In the maps, its pixel color is determined based on the traffic speed in its corresponding road segment.
- (2) We construct a stacked convolutional autoencoder to train a series of real-time traffic condition map. This article first explores the latent space in citywide traffic conditions. To measure the dynamics during peak hours, we compute the cosine similarities of the samples in a given period and study their statistical characteristics.

The organization of this paper is as follows. In Section 2, we conduct a literature review and define our problem. We introduce our method in Section 3 and then experiment to analyze the impact of Chongqing traffic restriction in 2018 in Section 4. After that, in Section 5, we compare the proposed method to the traditional histogram method and analyze our results. Finally, we end this article with a conclusion in Section 6.

2. Literature Review and Problem Definition

In this section, we conduct a brief review of traffic spatio-temporal-related research works and then make a definition of our problem.

2.1. A Brief Review of Related Works. Studies related to traditional traffic patterns based on physical laws and statistical methods are typically subject to single intersections, arterial roads, or small networks. From local to large-scale, urban traffic patterns are studied based on kinds of data. In the beginning, most researchers quantify the local traffic phenomenon based on traffic flow data collected using fixed-pointed detectors. Jin et al. [16] detect abnormal traffic with a robust principal component analysis. Treiber and Kesting [12] calibrate and validate traffic evolution by quantifying the bottleneck strength, modeling the propagation velocity, and checking the growth rate based on speed time series collected by aggregated detector data of several freeways in German. As we know, the deployment and maintenance of detectors are of high cost and are discrete. The data from them are not strictly continuous. With the development of communication and location technologies, data from communication and social media have become powerful supplements to fixed-pointed sensor data. These data expand the geography scope of the study. Using floating car data, He and Zheng [23] construct a spatiotemporal diagram to visualize traffic dynamics; Kartika [24] visualizes traffic congestion patterns; and Abinav et al. [25] estimate traffic state. Via vehicular ad hoc network (VANET), traffic congestion is identified based on event-driven architectures [26] or statistical network tomography [27]. However, such data have limited access. By the way, studies based on floating car data are depending on the trajectory of cars, which means that it cannot cover the whole city instantly. In most studied scenarios, urban expressways [28, 29] or road segment combinations [1, 30] stand as the large-scale area. They are only a part of the city. In 2019, Song et al. [31] integrate real-time traffic data retrieved from an online map and data from geographical detectors to mine the potential factors for each spatiotemporal pattern. The data from an online map are open, cheaply accessible, and practically validated, which can be an optional data source to push our research further for insight into patterns of traffic dynamics.

Strictly, the digital maps are provided based on traffic flow data. Based on traffic flow data, traffic conditions are mapped into different kinds of images to present citywide traffic spatiotemporal characteristics. Ma et al. [32] convert speed instantly to an image according to cars' positions. Based on a series of images, the authors predict a network-wide speed situation with high accuracy. Once the trajectory data are projecting to geographic maps, He and Zheng [23] construct spatiotemporal diagrams to show regional speed characteristics by the timeline of a day. From such diagrams, we can analyze the traffic condition's differences in regions and time. They also propose a simple mapping-to-cells method to visualize traffic dynamics based on floating car data. Once the traffic flow is determined, its value determines the color category in a geography map, which is the foundation for online digital maps. At present, digital map service providers assign different colors to the roads in the network on the basis of instant traffic conditions on them. To evaluate and express traffic conditions, various indexes are created. Furtlehner et al. [33] use travel time to generate local traffic indexes. These indexes will show excellent performance for the measurements in local regions. Digital maps can present these local regions in different colors according to their indexes. Usually, congested, slow, and smooth are colored, respectively, as red, orange, and green [6]. Digital maps keep the traffic spatial information in nature, and their updating can provide continuous temporal information. Digital map providers, such as Google Maps and Bing Maps, integrate multisource traffic data [31] and generate real-time and citywide traffic condition maps continuously [14]. The maps keep the spatial and temporal features [14, 34] for citywide traffic. These data are embedded in real navigation systems to support online traffic services, for example, route planning. They are public and real-time and maintain spatiotemporal information. Song et al. [31] analyze the congestion problem with the traffic data retrieved from such an online map. Lanet al. [35] provide the roads' temporal characteristics with speed data extracted from an E-map. Gong et al. [36] extract traffic status data from traffic condition maps provided by Baidu Maps to analyze large-scale congestion. Not only by these academic studies but also by practices, it can be seen that traffic index data or color information in digital maps are valid. They shall be a kind of well-integrated traffic data. However, to capture color features in a digital map precisely is a real challenge. There are kinds of ways to express color features in an image, from histograms to neural networks. A histogram is the most basic method that presents the distribution of the composition of colors. It shows both the types of colors and the number of pixels in each color that appeared. A histogram shall include three color histograms, which shows the distribution of the red/green/blue color channel. Li and Xiao [15] construct traffic index cloud maps (TICMs) based on traffic index value from a digital map, and they classify the city traffic states into two patterns, peak and nonpeak, based on histogram features in TICMs in the red channel. However, the histogram is only a statistical method that cannot reflect the pixels' relationship. Recently, CNNs apply to extract color features in images [32, 37, 38]. They can reveal correlations

between pixels. Intuitively, to a digital map, we can express it by a matrix or a long vector. It will be sparse and multiple for large-scale images. To show features vividly and clearly, scholars always consider transforming the matrix or long vector into a low-dimensional vector.

Besides, there are kinds of methodologies to transform an image into a low-dimensional vector. Due to the vast and complex data, they usually use feature reduction methods to extract the characteristics of those data and express them in a low-dimensional space. With the principal component analysis, Qu et al. [39] compress the network's flow volume data and Asif et al. [17] predict spatiotemporal patterns for large road networks. Zhang et al. [30] employ a dictionary-based compression theory to identify traffic patterns by analyzing the multidimensional traffic-related data. Yang et al. [28] reveal the heterogeneity of network traffic with a combination method of nonnegative tensor decomposition and clustering. These studies do not care about the time when these data are collected. During peak hours, the traffic conditions usually evolve tiny. Thus, their representing vectors will be sparse and similar. They need to be understood by some more effective methods. In another perspective, since traffic data in peak hours are related to commuter behaviors, its characteristics are recurrent [29] on specific road segments at a particular time [1]. As a result, for a predefined period, the features will be similar. Boquet et al. [20] prove that there is a low-dimensional latent space containing the underlying characteristics of the traffic. However, to the best of our knowledge, previous research studies do not investigate such latent space.

Recently, deep learning neural networks are popular in feature extraction and complex data representation. A stacked autoencoder can learn generic traffic flow features, which is the first time that a deep architecture model applies in representing such features [40]. A recurrent convolutional neural network [6] models the nonlinear relationship of adjacent road traffic. An extended deep belief network (DBN) [41] can do better exploitation in data with high nonlinearities and strong correlations. A stacked autoencoder [42] is proven to be effective in feature extraction in some constraints. These neural networks perform more favorably compared with some machine learning models. In particular, deep convolutional autoencoders are applied to extract a feature from pixel level from images. They have succeeded in learning synthetic aperture radar images [43, 44], computed tomography results [45], and electrocardiograms [46]. There, a convolutional layer extracts texture features, and an autoencoder's structure ensures the features' quality. The autoencoder, proposed by Hinton et al. as a generative model [47], shows excellent performance in nonlinear manifold learning [6]. This neural model projects data from high dimension to low dimension [48] and makes sure the extracted features are robust even in learning multiple modalities [49]. This method shall be suitable to capture traffic features in real-time traffic condition maps.

2.2. Problem Definition. We study those traffic condition maps to reveal the pattern evolution in the citywide traffic network.

Suppose traffic patterns are recurrent and a digital map provider updates its urban traffic condition maps timely and

accurately. Screenshots of such maps are timely and effective with the size $w \times h$. Each screenshot is treated as an instance X_i , where i records its generated moment. Taking each pixel p_m ($m \in [1, w \times h]$) as the feature dimension, we characterize the instance with its colors. We enumerate the color set $\text{color} = \{\text{red, yellow, green}\}$ as $\text{color} = \{2, 1, 0\}$. The color of each pixel on the urban traffic network is one of the three options. Thus, each real-time map implies citywide traffic conditions in pixels' color of its road network, expressed as a long and sparse vector $X_i = \{p_{1,i}, p_{2,i}, \dots, p_{w \times h,i}\}$, where $p_{m,i} \in \text{colour}$. For each screenshot, we extract the low-dimensional feature vector standing for the citywide traffic state at its generated moment. For a series of screenshots, there will be a series of low-dimensional feature vectors accordingly. To measure the dynamic of traffic conditions, we compute the distance between the series vectors. Here, we use these distances to identify the traffic patterns of the target area. We check the statistic characteristics of these distances' distribution to support traffic management. Hence, we define our problem as a combination of feature extraction and statistical measurement.

It is evident from the above literature review that urban traffic states are continuously expressed by digital traffic condition maps, which are introduced as a data source to provide traffic spatiotemporal information. Traditionally, related research data are from loop detectors, floating cars, or communication network. They are local, expensive, and not sure to be continuous. We use digital traffic condition maps as the research data, which is open and cheap and covers the spatiotemporal multi-scaled geography area. As of now, there has been a considerable amount of literature on the spatiotemporal correlation of citywide traffic conditions. However, there is still a lack of studies on the latent space hidden behind the traffic condition variant. This paper applies a stacked convolutional autoencoder to learn such space features through real-time traffic condition maps. This autoencoder differentiates traffic conditions combined spatial and temporal during peak hours. The vectors in the latent space can denote the traffic pattern evolution in a given period.

3. Data and Methodology

The color change in these real-time traffic condition maps records corresponding changes in road segment traffic conditions. Therefore, we propose a stacked convolutional autoencoder to capture the spatial distribution of the colors. After a brief introduction to our data, we will introduce the methodology in detail.

3.1. Data Description. Our input data are a time series of digital traffic condition maps, in which different colors for road segments illustrate their corresponding traffic conditions. Being common knowledge, red means congestion, yellow denotes low speed, and green signifies smooth flow. The purpose of the study uniformly determines the scale level of these digital maps. We only keep those pixels for road networks and trace their changes continuously for a

fixed time interval. Take Baidu Maps as an example. It will update digital maps every 3 minutes. The snapshot example, shown in Figure 1(a), is a snapshot of a real-time traffic condition map of Chongqing's main urban area. Figure 1(b) is the filtered urban road network structure for Figure 1(a), which only keeps those road network pixels. The default color of these road network pixels is green.

3.2. Methodology. We illustrate our proposed method in Figure 2, in which the original input is a series of real-time traffic condition map for a given period. There is a pre-process to keep road network pixels left only to be the input of our methodology.

Two terms are defined below before explaining the details of our proposed method.

Definition 1. naked road network traffic condition map (NRTM). Suppose a screenshot of a real-time traffic condition map has the size of w pixels \times h pixels with 3 red/green/blue channels. Let $U = \{p_{0,0}, p_{0,1}, \dots, p_{i,j}\}$, $i \in [0, w]$, $j \in [0, h]$ be its pixel set. There exists a subset $U' = \{p'_{m,n}\}$, $m \in [0, w]$, $n \in [0, n]$, whose elements are on the road network. The color of pixel $p'_{m,n}$ is $C_{p'_{m,n}} \in \{\text{red, green, blue}\}$. These pixels in U' compose NRTM.

Definition 2. Traffic condition pattern during peak hours. Each NRTM has a meaningful vector expressing global traffic conditions at its peak moment t . Given t_0 is the beginning moment; Δt is a fixed time interval; X_i is the i th NRTM ($t_i = t_0 + i \times \Delta t$); X_j is the j th NRTM ($t_j = t_0 + j \times \Delta t$); the function f maps each NRTM to a feature vector, that is, to X_i , its feature vector is $\widehat{X}_i = f(X_i)$; to X_j , its feature vector is $\widehat{X}_j = f(X_j)$; and the distance $\text{distance}_{i,j}$ illustrates the similarity between these two NRTMs, which can show their difference. Once there are series of such continuous distances, it can trace the variant of traffic conditions. We use the statistical characteristics of this series to describe the traffic condition pattern in peak hours.

A series of NRTM is the input of our methodology. Our target is to find traffic condition pattern in peak hours. There are three steps to find different traffic condition patterns. First, a specially designed convolutional autoencoder extracts a 3D feature vector for each image. Then, for a given period, the distances between vectors are calculated by their cosine similarity. Finally, different patterns will be revealed for different periods according to the statistic characteristics of these distances.

3.2.1. Extract Latent Features from Digital Maps. In Figure 2, the core component is a stacked convolutional autoencoder. It transfers an image into a vector according to its RGB color spatial distribution.

Autoencoder is a symmetrical-structural self-supervised neural network based on an encoder and a decoder. When convolutional layers substitute fully connected layers in an autoencoder, a convolutional autoencoder is constructed. It

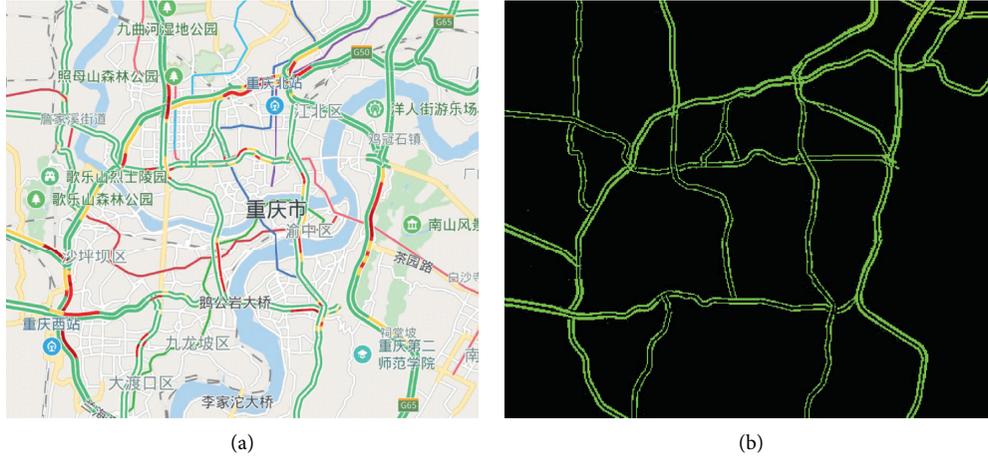


FIGURE 1: Sample of our input data. (a) the raw screenshot of Baidu Maps. (b) the filtered traffic condition map based on the citywide traffic network.

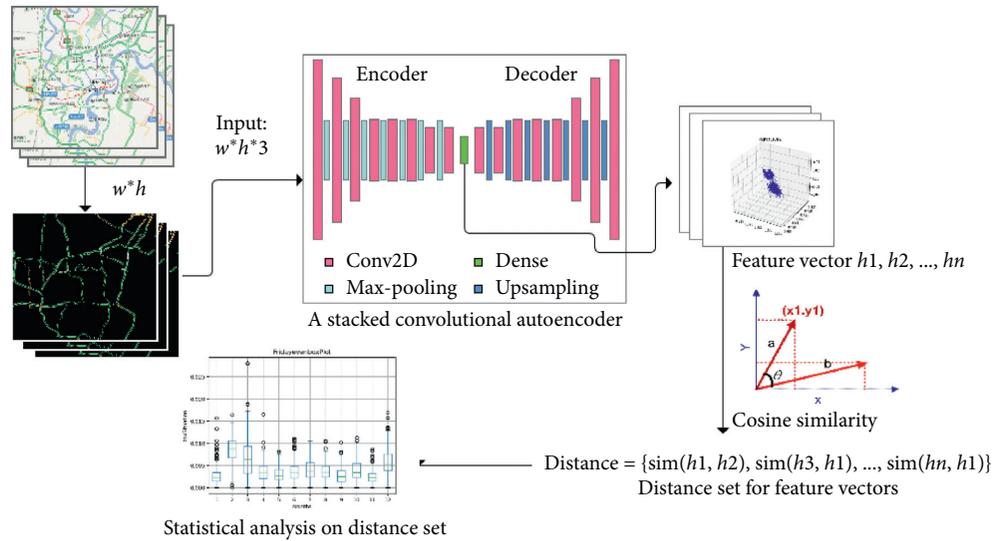


FIGURE 2: Our proposed method framework. The input is a series of NRTM with the size of $w \times h \times 3$. They are fed to a stacked convolutional autoencoder and transformed into a series of three-dimensional feature vectors. The distances between vectors are measured by cosine similarity. Then, the statistical analysis describes their distribution characteristics.

captures and represents the low-dimensional features of our input at its hidden central layer.

In terms of a convolutional neural network unit, it consists of a convolution layer and a pooling layer [50]. In the convolution layer, filters are convolved with the receptive field of the input image in a sliding window to learn data-specific features. The dimension of its sliding sub-window determines the learning granularity. To get the pixel-level feature in RGB mode, we set the dimension as $k \times k$ with strides s . For the pooling layer, we choose $m \times m$ max-pooling computing to capture upper limits in subwindows. Generally, a pooling layer follows each convolutional layer. Max-pooling is basically a nonlinear downsampling procedure, which helps to reduce the computational complexity for the forward layers, as well as adding translation invariance to the network.

For a given input vector of X , an autoencoder tries to approximate $si(X) \cong X$. The function si is the target function the neural network learns, which captures features in the latent space of input data and then reconstructs them. The loss between X and $si(X)$ will be used in the back-propagation algorithm to ensure the accuracy of the network. Once the network is well-trained, based on its structure parameters, we get a low-dimensional feature vector. The encoder computes a nonlinear mapping of the inputs as follows:

$$h_i = f(W_e x_i + b_e), \quad (1)$$

where W_e and b_e denote weights and bias of the encoder, respectively.

The decoder reconstructs the input as follows:

$$si(x_i) = f(W_d h_i + b_d), \quad (2)$$

where W_d and b_d denote weights and biases of the decoder, respectively. They also belong to the parameter sets of the autoencoder. The dimension of the output layer equals the size of h , the hidden central layer.

During unsupervised pre-training, the network tries to minimize the reconstruction error

$$J(\theta) = \frac{1}{N} \sum_{i=1}^N (x_i - si(x_i))^2. \quad (3)$$

by tuning its weights and biases. During tuning, mean square error is selected as the loss function.

With the stacked convolutional autoencoder shown in Figure 3, we transform an image into a low-dimensional feature vector h . Figure 4 shows the illustrative architecture of our proposed stacked convolutional autoencoder. In the decoder, we choose the combination of convolutional layer and pooling layer. Convolutional autoencoders combine the benefits of convolutional filtering in CNN's with unsupervised pretraining of autoencoders. The encoder contains convolutional layers, and the decoder contains deconvolutional layers. Each deconvolutional layer shall be followed by an unpooling layer. The unpooling operation is performed by storing the locations of the maximum values during pooling, preserving the values of the locations during unpooling and zeroing the rest. For a set of images in a particular period, there will correspondingly be a set of uniform, low-dimensional vectors. We set the layer number according to the characteristics of inputs, the dimension of outputs, and the computation of convolutional layers. In terms of training performance, we choose the proper kernel of the convolutional layer, the loss measurement, and the optimizer.

3.2.2. Compute Similarities between Feature Vectors during a Given Period. In this section, we investigate the distribution characteristics of the low-dimensional feature vectors during an observation period. Referring to measurement in social networks [51]; the values of cosine similarity between these vectors are checked.

In a given period, there are N images that are denoted as set A . Correspondingly, there are N low-dimensional vectors. We describe them as $\widehat{H}_A = [h_{A1}, h_{A2}, \dots, h_{AN}]$, where h_{Ai} is the feature vector for an image generated at time i .

As we know, the traffic conditions' evolution at peak times is tiny, that is, these vectors are very closely mapping into a latent space and the distances between these vectors are small. We calculate the cosine similarity between h_{Ai} and h_{Aj} according to

$$\text{Sim}(h_{Ai}, h_{Aj}) = \frac{h_{Ai} \cdot h_{Aj}}{|h_{Ai}| \times |h_{Aj}|}, \quad (4)$$

where $i, j \in [1, N]$. Thus, a similarity matrix for A is obtained.

Here, the similarity between two adjacent screenshots is used to measure the variant of traffic patterns. We will check

the statistical characteristics between feature vectors captured as the feature variant of traffic patterns.

4. Experiment and Results

We apply our method to investigate the impact of a vehicle restriction policy on the urban traffic condition in 2018 in Chongqing. To our target, we category these screenshots into four classes: morning and evening peak hours and in and out of the restriction period.

4.1. Data Set. Chongqing is one of the metropolitan areas in China. There are two rivers named Yangtze River and Jialing River going through Chongqing's main city area. Bridges play a crucial role in the traffic system. Between April 21 and November 7, 2018, the transportation department maintains three bridges named Huanghuayuan, Jiahua, and Yuao, which connect three districts Yuzhong, Jiangbei, and Nanan. Figure 5 shows the locations of these three bridges. Taking into account the load-bearing safety of bridges and the massive traffic demand for crossing the river during working days, the government implemented a one-day-per-week vehicle restriction strategy from 7 am to 10 pm during the maintenance. We analyze the changes in the traffic spatiotemporal pattern because of the implementation of this restriction policy.

We obtain the raw snapshots of traffic condition maps that are updated by Baidu Maps every 3 minutes. The map we use is in level 12, shown in Figure 1(a), covering about 700 km² area of Chongqing's central city. We extract the arterial road network out, shown as Figure 1(b), omitting many disturbance pixels, as the input of our method. There are 50,688 useful snapshots on workdays from January 1 to December 31 in 2018. In detail, there are 25,102 screenshots in the restricted period and 25,586 ones out of that period. We set the morning peak time from 7 am to 10 am and the evening peak time from 5 pm to 8 pm. In terms of different peak hours, there are 20,734 snapshots for the morning peak and 29,954 for the evening peak. We conduct experiments with these four divided data sets.

4.2. Parameters of the Stacked Convolutional Autoencoder. To achieve better training performance, we set the stacked convolutional autoencoder shown in Figure 3.

Our inputs cover Chongqing's main urban area with a size of $w = h = 500$. We divide the data into two subsets, namely, the training set and validation set. The validation split ratio is 0.3, which is a kind of normal setting in machine learning. We set h as a three-dimensional vector. If we set it as only one dimension, we get only a line for a time series. Otherwise, if it is a two-dimensional vector, we get a flat. To represent these vectors in a stereo coordinate, we set it as a three-dimensional vector.

We choose the hyperbolic tangent function $f(z) = \tanh(z) = (e^z - e^{-z}) / (e^z + e^{-z})$ as the activation function, whose derivative is related to $\tanh(z)$ itself, and there is a property of $\tanh'(z) = 1 - \tanh^2(z)$.

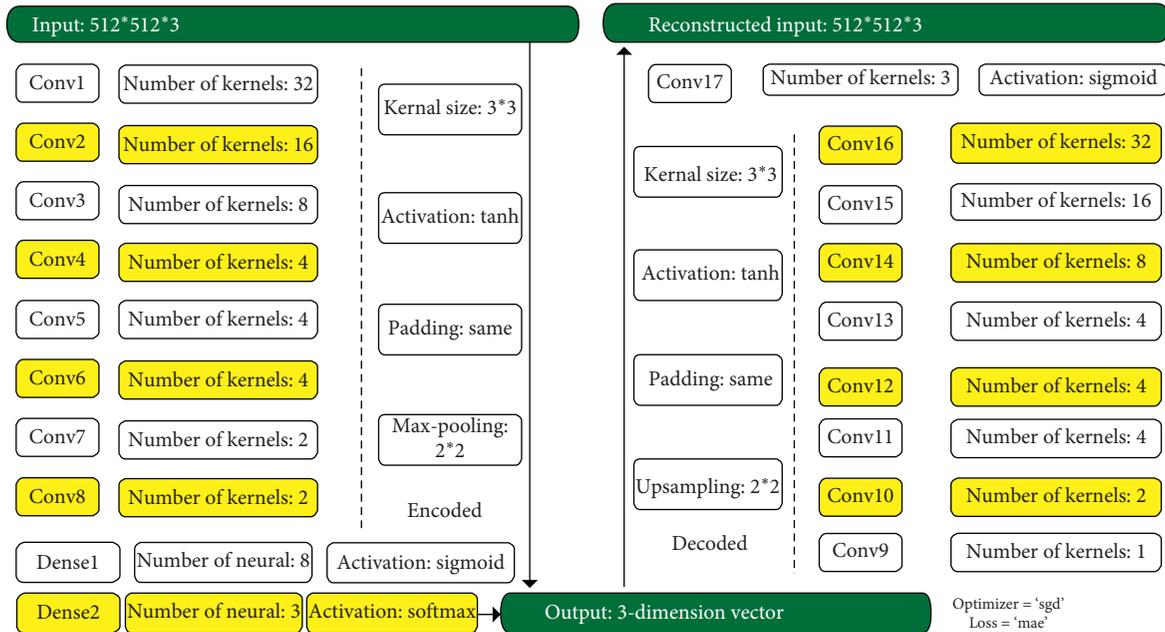


FIGURE 3: Parameters of the proposed stacked convolutional autoencoder. There is a fully connected layer with eight neurons and a softmax classifier with three neurons added at the end of the encoder. For the eight convolution layers in the encoder and the convolution layers from 9 to 16 in the decoder, kernel size, activation function, padding, and max-pooling are the same, only the number of kernels is changing, shown in the encoded and decoded part, respectively. For the convolutional layer 17, the number of kernels is 3, and the activation function is sigmoid.

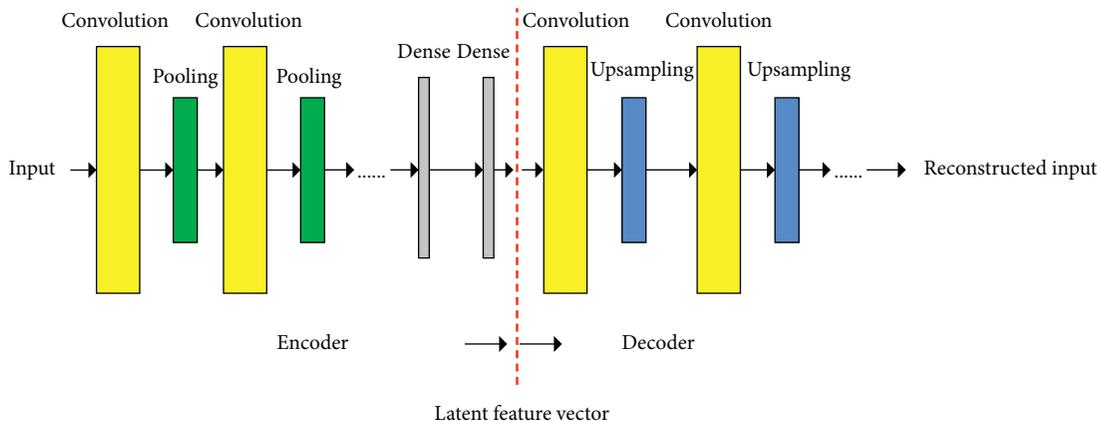


FIGURE 4: Illustrative architecture of the stacked convolutional autoencoder.

The input is a series of RGB images consisting of three channels. The value range for each channel is $[0, 255]$. Since a convolutional neural network performs better for data ranging from 0 to 1, we normalize our input. Suppose B is the original array for an image whose size is $n \times m$, $B(i, j)$ is the original pixel color value at position (i, j) , where $i \in [1, n]$ and $j \in [1, m]$. We set $B'(i, j) = B(i, j)/255$ to get the normalized input B' .

There are 17 convolutional layers in this special autoencoder. As we know, traffic conditions have the property of propagation for the adjacent area [52, 53], so there will be a relationship between some adjacent pixels. To keep such features in pixels, we set $k = 3$, $s = 1$, $m = 2$ for convolution layers, that is, we use a 3×3 subwindow to

finish the spatial feature capturing with stride 1. Subsequently, we choose a 2×2 max-pooling filter [54] to get information at the pixel level with the same padding. We select mean absolute error (MAE) to measure the loss, which reflects the situation of predicted error directly. We choose the stochastic gradient descent (SGD) to be the optimizer with a self-adaptive learning rate [55]. We monitor the value of loss to ease the gradient vanishing problem and set the minor learning rate as $lr = 0.001$.

4.3. *Training Performance of the Stacked Convolutional Autoencoder.* We use loss curves to show the training performance of our proposed stacked convolutional

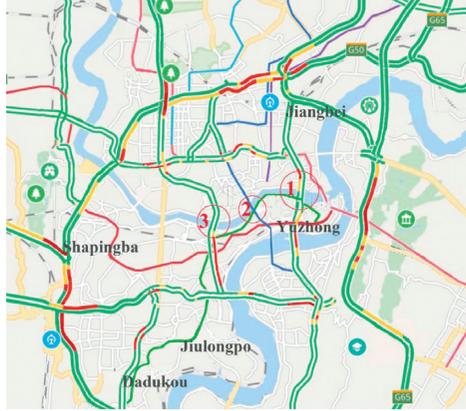


FIGURE 5: Locations of bridges in maintaining: locations of bridges Huanghuayuan in position 1, Yuao in position 2, and Jiahua in position 3.

autoencoder. Four classified images are fed to the model, respectively. Figure 6 illustrates the loss curves for the two-class data during morning peak hours. It is evident that the loss of the stacked convolutional autoencoder starts from a small value and converges quickly. The result demonstrates that our parameters in the neural network make it converge fast without the gradient vanishing problem and learn the data efficiently.

4.4. Traffic Patterns Change Caused by Restriction. Theoretically, the restriction policy shall mainly affect the traffic volume in and out of the Yuzhong district, which is the traditional urban function area. During those days without restriction, these volumes are scattered on the three bridges. Once the restriction policy is executed, without the support of these three bridges, the volumes may be concentrated on left roads linking to the Yuzhong district. There shall be patterns change by comparing them in and out of restriction.

To find changes in the same peak hours of and outside the transportation restriction period, we represent the feature vectors of latent space with scatter plot of the same coordinate system, illustrating with Figure 7. Observing (a) and (b) in Figure 7, we can find that because of the restriction policy, both the morning and evening peak patterns have changed. Relatively, the vectors will be denser than during the time period without restriction.

5. Comparison and Analysis

In this section, we design experiments to validate the meaning and feasibility of our proposed method. It is the first time to provide insight into the citywide traffic pattern evolution based on colors in digital maps. As we know, the histogram is the most famous method to study colors in images. Li and Xiao [15] have applied it to detect traffic peak periods by color vectors of their TICMs. In practice, we also determine the citywide traffic conditions according to the number of red and yellow pixels. Since the histogram is essentially a statistical analysis, we use it to validate our results. We compare the feature extraction performance between the histogram and our proposed convolutional

autoencoder. Furthermore, based on the vectors we obtained, we analyze the pattern change in weekdays, vacation days, and different peak times in the morning and evening.

5.1. Richer Feature Extracted Compared to Histogram. A histogram is a primary method to calculate the number of different color pixels. In [15], the histogram feature vectors classify TICMs into peak and nonpeak categories according to the pixels' color distribution feature. Here, we compare our method with a histogram to show its performance in extracting features of these images. We try to use histogram features to identify patterns of our image data during peak hours. The results of the red channel are shown in semi-log plots, illustrating as Figure 8. It can identify the distribution of different level red pixels. In Figure 8, we find two points. First, in the restriction period, the number of light red pixels is smaller than that out of the restriction period. Second, restriction policy has little influence on the distribution of red pixels valued between 50 and 255. Based on the two findings, we conclude that the feature extracted by histogram can hardly distinguish the peak traffic states in restrictions from that of out of restrictions. However, with the stacked convolutional autoencoder, the distribution difference is noticeably shown in Figure 7. That is, our autoencoder captures more detailed features than the histogram. After all, the histogram cannot find the gradient between pixels while our stacked convolutional autoencoder can.

5.2. Different Traffic Pattern Evolutions in Workdays. Our input data cover the morning peak time for eight months and the evening peak time for twelve months. The policy is only active on workdays. We try to provide insight into the traffic pattern evolution by comparing their statistical characteristics in and out of the restriction. Box diagrams show these characteristics. We provide rectangles for the periods with restriction in red in Figure 9.

From Figure 9, it is obvious that the distance between vectors in the morning peak is relatively stable except on Friday. In the restriction period from May to August, the average level and the variance level of distance are both lower than that from January to April. That is, because of the

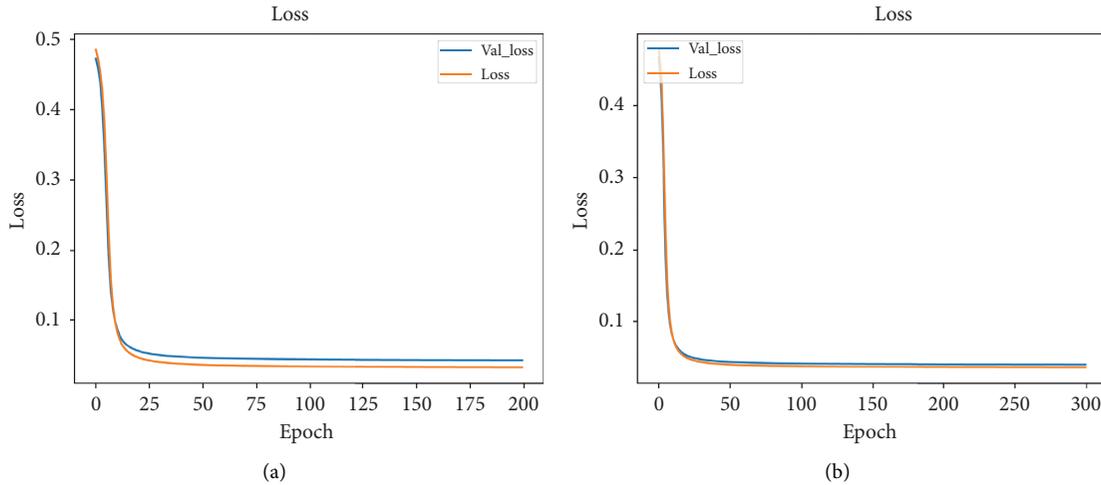


FIGURE 6: Training loss for data without and within restriction periods. (a) The loss for training workday morning data without restriction. (b) The loss for training workday morning data with restriction.

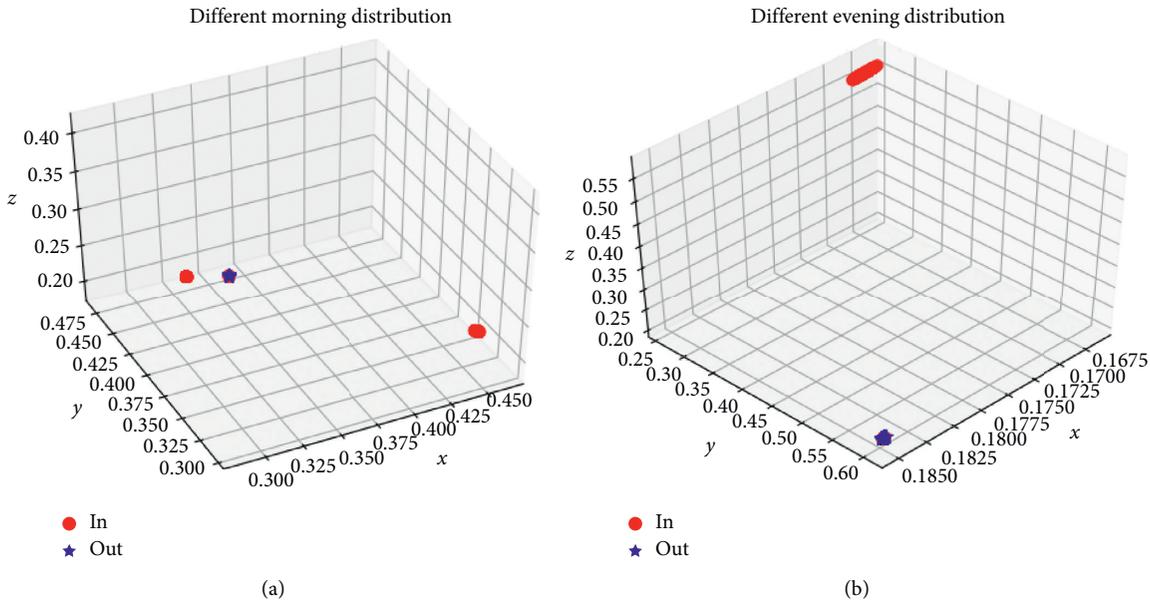


FIGURE 7: Vectors captured by the proposed stacked convolutional autoencoder. The latent space features are three dimensions. Vectors are divided into four categories, morning and evening peak hours and in and out of the restriction period. Scatter plots are organized by peak hours: (a) morning peak hours and (b) evening peak hours. The red circle marker is vector in the restriction period, and the blue star marker is vector out of the restriction period.

restriction and the common commuter demand, with the limited route options, traffic conditions are denser in the restriction period than out of the restriction period.

However, the situation in the evening peak is different. For Monday, it is more fluctuated in May and June than in other months. Then, the traffic condition seems to be stable. For Tuesday, the evening peak traffic is the most dynamic in a week according to the variance of the distances. The evening peak traffic on Wednesday is the most stable one.

5.3. Recurrent Traffic Patterns during Holidays. Since the restriction policy is not active on vacation days, we check the

results of our method on these days to validate the opposite. We category holidays and weekends as vacation days and compare between the months as shown in Figure 10. There are no significant differences between in and out of the restriction period. It is reasonable because the restriction policy is only for workdays. Such a phenomenon validates the feasibility of our proposed method from the other side.

5.4. Evolution of Different Traffic Patterns in Morning and Evening Peak Hours. Even traffic conditions are similar during peak hours, gradual changes exist. We try to provide the distance curves to find such evolution. Figure 11 shows changes in

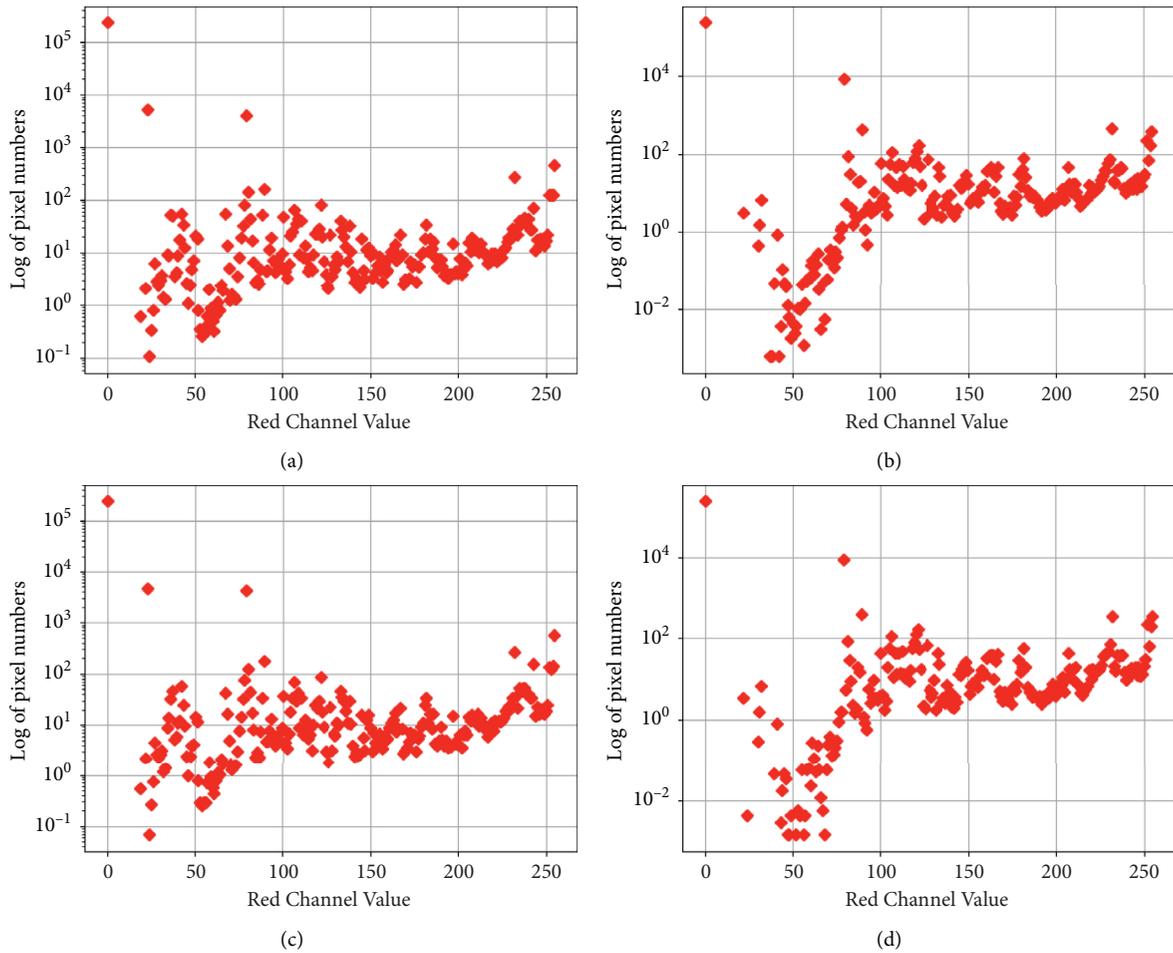


FIGURE 8: Feature in red channel captured by the traditional histogram method: the horizontal axis Red Channel Value is the color value of red channel, Red Channel Value $\in [0, 255]$; the vertical axis is the logarithm calculation of the average number of pixels: (a) logarithm calculation of average number of pixels in red channel in morning peak hours out of restriction period, (b) logarithm calculation of average number of pixels in red channel in morning peak hours in restriction period, (c) logarithm calculation of average number of pixels in red channel in evening peak hours out of restriction period, and (d) logarithm calculation of average number of pixels in red channel in evening peak hours in restriction period.

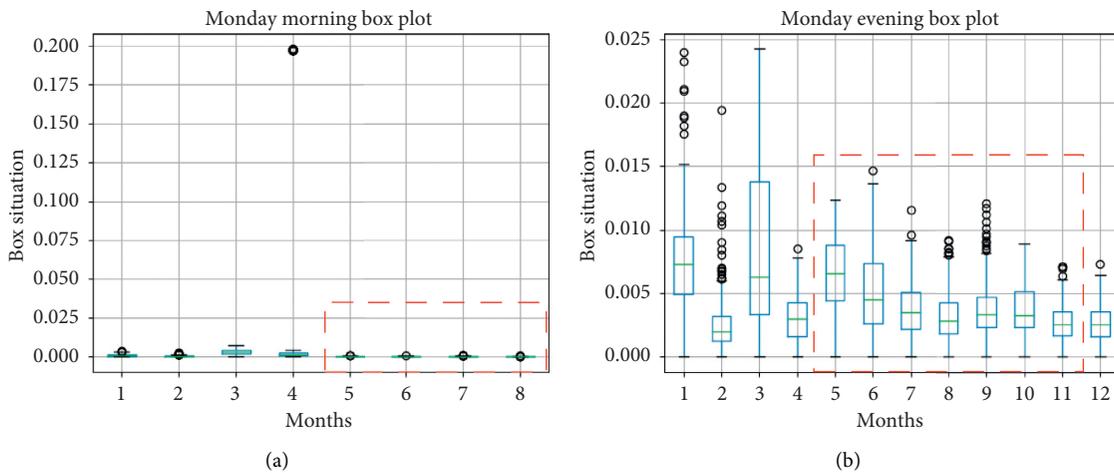


FIGURE 9: Continued.

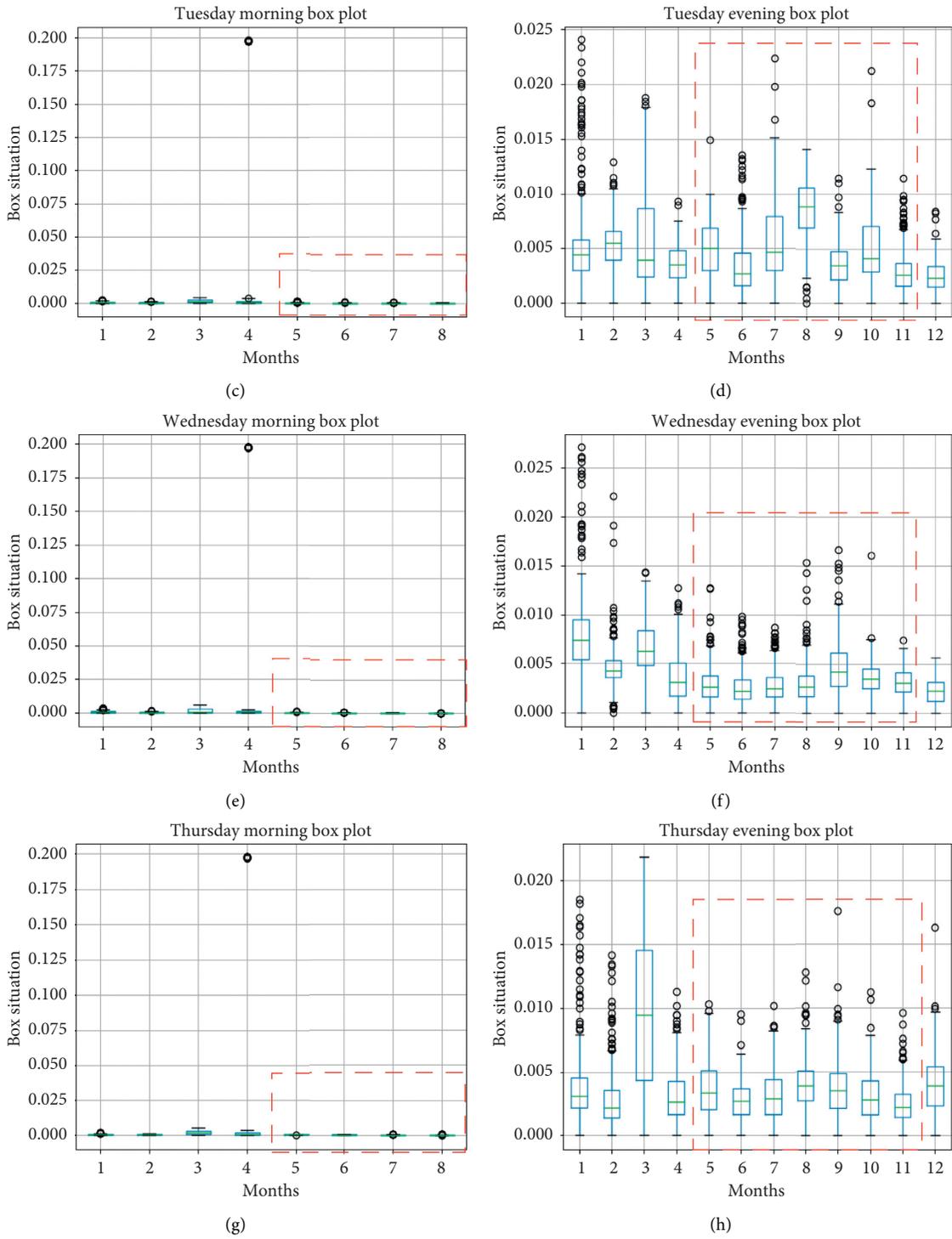


FIGURE 9: Continued.

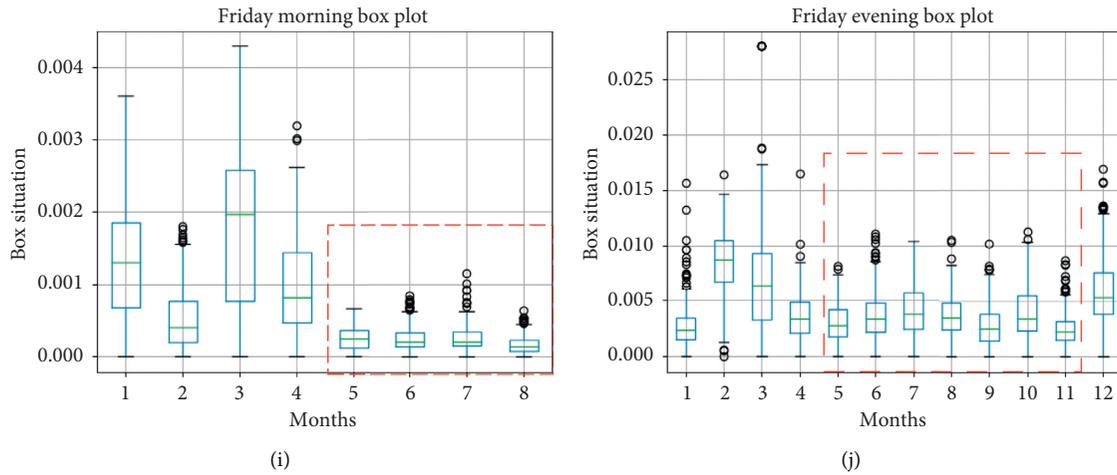


FIGURE 9: Box diagrams showing different distributions of vector distance in workdays in different months. The horizontal axis represents the months. The vertical axis represents the statistical status of vector distance: (a) distribution of distances in Monday morning peak, (b) distribution of distances in Monday evening peak, (c) distribution of distances in Tuesday morning peak, (d) distribution of distances in Tuesday evening peak, (e) distribution of distances in Wednesday morning peak, (f) distribution of distances in Wednesday evening peak, (g) distribution of distances in Thursday morning peak, (h) distribution of distances in Thursday evening peak, (i) distribution of distances in Friday morning peak, and (j) distribution of distances in Friday evening peak.

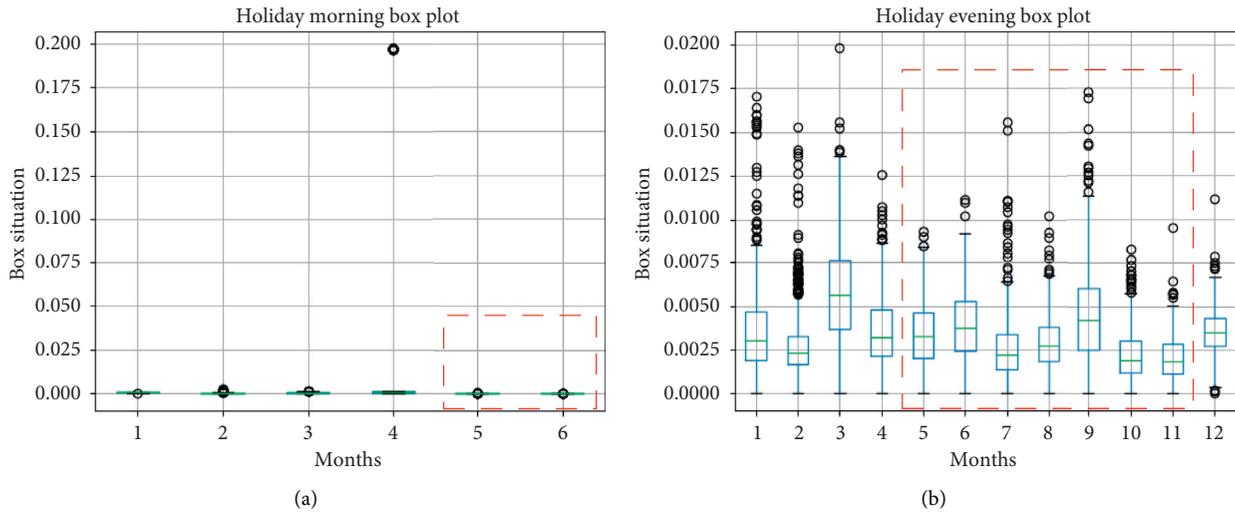


FIGURE 10: Box diagrams showing a different distribution of distances on vacation days in different months. The horizontal axis represents the months. The vertical axis represents the statistical status of distances. (a) Distribution of distances in morning peak on vacation days and (b) distribution of distances in evening peak on vacation days.

the morning and evening peak times. It illustrates there is a steep change point both in the morning and evening peak hours, which happens at the beginning of the restriction policy. From the overall trend, the distance in the morning fluctuates more

than that in the evening. But in detail, the change range in the evening is more frequent than that in the morning.

We can track such changes in a long time to represent the evolution of traffic conditions at peak hours.

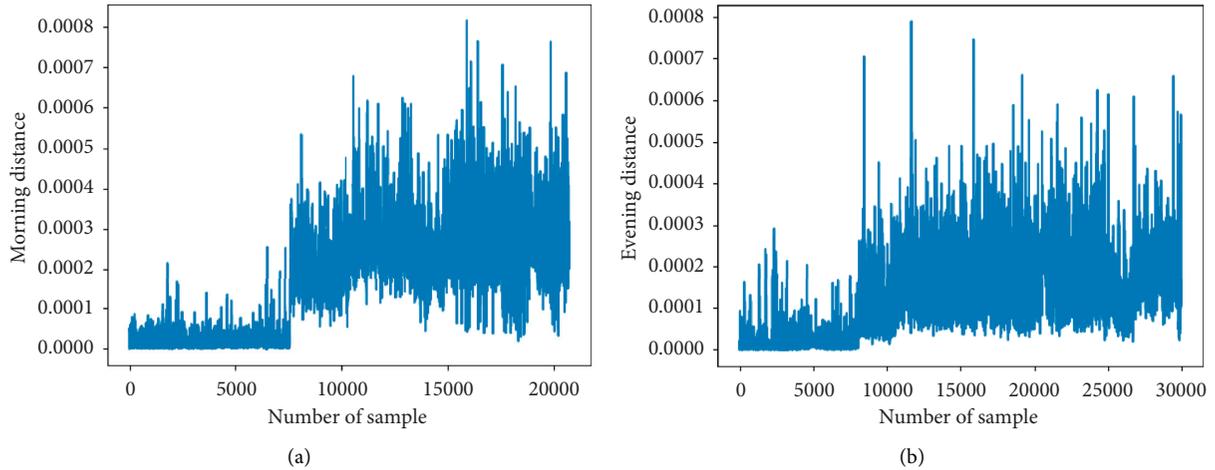


FIGURE 11: Distance change during peak times: take the first vector in morning and evening peak time as the baseline. The horizontal axis represents the number of samples (a time series). The vertical axis represents the distance between vectors. (a) Distance evolution during morning peak times and (b) distance evolution during evening peak times.

6. Conclusion and Discussion

6.1. Conclusion. In this paper, we have proposed a convolutional autoencoder-based framework to identify the dynamics of citywide traffic conditions. Considering the accessibility and geographical coverage of data, we choose real-time, traffic condition digital maps as our inputs. Digital maps label roads' traffic states with different colors. The change of pixel color is the mapping of the evolution of road traffic conditions. Our stacked convolutional autoencoder extracts latent representation vectors from each map. For a series of maps in a given period, we get a series of vectors. Distances between vectors denote traffic pattern evolution. We have evaluated our framework on the map of Baidu traffic conditions in Chongqing city to estimate the impact of a vehicle restriction policy and achieve performance that is significantly beyond the traditional histogram method. We find that there are two different impacts in the morning and the evening peak times. The evolution is tiny but continuous. From the long-term perspective, the restriction policy is pushing the evolution of peak-time traffic.

6.2. Managerial Insights. Different from traditional traffic pattern studies, we are concerning the changes of citywide traffic conditions to not only identify and represent the traffic conditions but also to understand more in traffic evolution. From the evidence of the case study, this paper offers two implications for managers:

- (i) From the result of Section 5.2, the traffic is more concentrated in morning peak time than that in evening peak time. Managers and operators can pay attention to those key congestion points in the morning to improve their management efficiency.
- (ii) According to the result of Section 5.4, we can find there are even changes both for morning and evening peak hours, but the trend in the morning is more fluctuating. Managers should pay more attention to traffic guidance in the morning.

6.3. Limitation and Future Research. The limitation of our method is twofold. First, autoencoder is a feature extraction method with information loss. It is difficult to explain the feature explicitly in detail. Second, the inputs are integrated data, which are the results of multiple social and natural factors. In the future, we can extend our work in two directions. (1) We will add information on weather, atmosphere, and social activities such as meetings and shows to reveal their relationship with the change in traffic conditions. (2) We will use our proposed method to study different cities to find some more generic features in urban transportation.

Data Availability

The pictures' data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This article was supported by the National Natural Science Foundation of China (Grant nos. 71871034, 71871035, 71471024, and 71701116).

References

- [1] X. L. Ma, L. Yi, and C. Peng, "Identifying spatiotemporal traffic patterns in large-scale urban road networks using a modified nonnegative matrix factorization algorithm," *Journal of Traffic and Transportation Engineering (English Edition)*, vol. 7, p. 002, 2018.
- [2] Y. Sun, M. Hrušovský, C. Zhang, and M. Lang, "A time-dependent fuzzy programming approach for the green multimodal routing problem with rail service capacity uncertainty and road traffic congestion," *Complexity*, vol. 2018, 22 pages, 2018.

- [3] J. Zhou, H. Chang, X. Cheng, and X. Zhao, "A multiscale and high-precision LSTM-GASVR short-term traffic flow prediction model," *Complexity*, vol. 2020, 17 pages, 2020.
- [4] J. Yuan, Y. Zheng, X. Xie, and G. Z. Sun, "T-drive: Enhancing driving directions with taxi drivers' intelligence," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 1, pp. 220–232, 2013.
- [5] Y. Su, X. Liu, and X. Li, "Research on traffic congestion based on system dynamics: the case of Chongqing, China," *Complexity*, vol. 2020, 13 pages, 2020.
- [6] Y. Zhu, X. Wu, P. Li, Y. Zhang, and X. Hu, "Transfer learning with deep manifold regularized auto-encoders," *Neurocomputing*, vol. 369, pp. 145–154, 2019.
- [7] F. Shi, G.-M. Xu, B. Liu et al., "Optimization method of alternate traffic restriction scheme based on elastic demand and mode choice behavior," *Transportation Research Part C: Emerging Technologies*, vol. 39, pp. 36–52, 2014.
- [8] X. Chen, Z. He, Y. Chen, Y. Lu, and J. Wang, "Missing traffic data imputation and pattern discovery with a Bayesian augmented tensor factorization model," *Transportation Research Part C: Emerging Technologies*, vol. 104, pp. 66–77, 2019.
- [9] I. G. Guardiola, T. Leon, and F. Mallor, "A functional approach to monitor and recognize patterns of daily traffic profiles," *Transportation Research Part B: Methodological*, vol. 65, pp. 119–136, 2014.
- [10] W. Shen and H. M. Zhang, "On the morning commute problem in a corridor network with multiple bottlenecks: its system-optimal traffic flow patterns and the realizing tolling scheme," *Transportation Research Part B: Methodological*, vol. 43, no. 3, pp. 267–284, 2009.
- [11] F. G. Habtemichael and M. Cetin, "Short-term traffic flow rate forecasting based on identifying similar traffic patterns," *Transportation Research Part C: Emerging Technologies*, vol. 66, pp. 61–78, 2016.
- [12] M. Treiber and A. Kesting, "Validation of traffic flow models with respect to the spatiotemporal evolution of congested traffic patterns," *Transportation Research Part C: Emerging Technologies*, vol. 21, no. 1, pp. 31–41, 2012.
- [13] K. Mariano and J. M. Ale, "Discovering traffic congestion through traffic flow patterns generated by moving object trajectories," *Computers, Environment and Urban Systems*, vol. 80, Article ID 101426, 2019.
- [14] S. Luís, C. R. João, and H. A. Carlos, "A web spatial decision support system for vehicle routing using Google maps," *Decision Support Systems*, vol. 51, no. 1, pp. 1–9, 2011.
- [15] Y. N. Li and J. L. Xiao, "Traffic peak period detection using traffic index cloud maps," *Physica A: Statistical Mechanics and its Applications*, vol. 553, Article ID 124277, 2020.
- [16] X. Jin, Y. Zhang, L. Li, and J. Hu, "Robust PCA-based abnormal traffic flow pattern isolation and loop detector fault detection," *Tsinghua Science and Technology*, vol. 13, no. 6, pp. 829–835, 2008.
- [17] M. T. Asif, J. Dauwels, C. Yang Goh et al., "Spatiotemporal patterns in large-scale traffic speed prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 2, pp. 794–804, 2014.
- [18] Y. Hernandez, "Local traffic patterns extraction with network-wide consistency in large urban networks," in *Proceedings of the International Symposium of Transport Simulation and the International Workshop on Traffic Data Collection and its Standardization*, Matsuyama, Japan, August 2018.
- [19] F. Rempe, G. Huber, and K. Bogenberger, "Spatio-temporal congestion patterns in urban traffic networks," *Transportation Research Procedia*, vol. 15, pp. 513–524, 2016.
- [20] G. Boquet, M. Antoni, S. Javier, and L. V. Jose, "A variational autoencoder solution for road traffic forecasting systems: missing data imputation, dimension reduction, model selection and anomaly detection," *Transportation Research Part C: Emerging Technologies*, vol. 115, Article ID 102622, 2020.
- [21] D. Michal, B. Xavier, and V. Pierre, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 3844–3852, Barcelona, Spain, December 2016.
- [22] H. Y. Fu, H. D. Ma, G. Y. Wang, X. M. Zhang, and Y. F. Zhang, "MCFF-CNN: multiscale comprehensive feature fusion convolutional neural network for vehicle color recognition based on residual learning," *Neurocomputing*, vol. 395, pp. 178–187, 2020.
- [23] Z. B. He and L. Zheng, "Visualizing traffic dynamics based on floating car data," *Journal of Transportation Engineering, Part A: Systems*, vol. 143, no. 5, Article ID 04017005, 2017.
- [24] C. Kartika, "Visual exploration of spatial-temporal traffic congestion patterns using floating car data," Master's thesis, Technische Universität München, Munich, Germany, 2015.
- [25] S. Abhinav, V. Viswanathan, W. T. Cai, and A. Knoll, "Traffic state estimation using floating car data," *Procedia Computer Science*, vol. 80, pp. 2008–2018, 2016.
- [26] T. S. Fernando, V. V. Mercedes, S. M. Cristina, T. M. Rafael, and F. G. S. Antonio, "A cooperative approach to traffic congestion detection with complex event processing and VANET," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 2, pp. 914–929, 2012.
- [27] A. Paranjothi, M. S. Khan, R. Patan, R. M. Parizi, and M. Atiqzaman, "VANETomo: a congestion identification and control scheme in connected vehicles using network tomography," *Computer Communications*, vol. 151, pp. 275–289, 2020.
- [28] S. Y. Yang, J. P. Wu, Y. Y. Xu, and T. Yang, "Revealing heterogeneous spatiotemporal traffic flow patterns of urban road network via tensor decomposition-based clustering approach," *Physica A: Statistical Mechanics and its Applications*, vol. 526, 2019.
- [29] Z. Zhang, Y. Wang, P. Chen, Z. He, and G. Yu, "Probe data-driven travel time forecasting for urban expressways by matching similar spatiotemporal traffic patterns," *Transportation Research Part C: Emerging Technologies*, vol. 85, pp. 476–493, 2017.
- [30] Z. Zhang, Q. He, H. Tong, J. Gou, and X. Li, "Spatial-temporal traffic flow pattern identification and anomaly detection with dictionary-based compression theory in a large-scale urban network," *Transportation Research Part C: Emerging Technologies*, vol. 71, pp. 284–302, 2016.
- [31] J. C. Song, C. L. Zhao, S. P. Zhong, T. A. S. Nielsen, and A. V. Prishchepov, "Mapping spatio-temporal patterns and detecting the factors of traffic congestion with multi-source data fusion and mining techniques," *Computers, Environment and Urban Systems*, vol. 77, Article ID 101364, 2019.
- [32] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. P. Wang, "Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction," *Sensors*, vol. 17, no. 4, pp. 818–827, 2017.
- [33] C. Furtlehner, Y. F. Han, J. M. Lasgouttes et al., "Spatial and temporal analysis of traffic states on large scale networks," in *Proceedings of the 2010 13th International IEEE Annual*

- Conference on Intelligent Transportation Systems*, Madeira Island, Portugal, September 2010.
- [34] N. Ferreira, J. POCO, H. T. Vo, J. Freire, and C. T. Silva, "Visual exploration of big spatio-temporal urban data: a study of New York city taxi trips," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2149–2158, 2013.
- [35] Z. Q. Lan, C. M. He, and M. Cai, "Urban road traffic noise spatiotemporal distribution mapping using multisource data," *Transportation Research Part D: Transport and Environment*, vol. 82, Article ID 102323, 2020.
- [36] K. Gong, L. Zhang, D. Ni et al., "An expert system to discover key congestion points for urban traffic," *Expert Systems with Applications*, vol. 158, pp. 1–15, Article ID 113544, 2020.
- [37] J. Jin, K. Fu, and C. Zhang, "Traffic sign recognition with hinge loss trained convolutional neural networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 1991–2000, Article ID 2308281, 2014.
- [38] Y. Lecun and S. Pierre, "Traffic sign recognition with multi-scale convolutional networks," in *Proceedings of the International Joint Conference on Neural Networks*, San Jose, CA, USA, August 2011.
- [39] L. Qu, J. M. Hu, and Y. Zhang, "A flow volumes data compression approach for traffic network based on principal component analysis," in *Proceedings of the Intelligent Transportation Systems Conference*, Seattle, WA, USA, September 2007.
- [40] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: a deep learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, pp. 1–9, Article ID 2345663, 2014.
- [41] Y. Wang, Z. Pan, X. Yuan, C. Yang, and W. Gui, "A novel deep learning based fault diagnosis approach for chemical process with extended deep belief network," *ISA Transactions*, vol. 96, pp. 457–467, 2020.
- [42] X. Yuan, J. Zhou, B. Huang, Y. Wang, C. Yang, and W. Gui, "Hierarchical quality-relevant feature representation for soft sensor modeling: a novel deep learning strategy," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 3721–3730, 2020.
- [43] J. Geng, J. Fan, H. Wang, X. Ma, B. Li, and F. Chen, "High-resolution SAR image classification via deep convolutional autoencoders," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 11, pp. 2351–2355, 2015.
- [44] R. H. Shang, G. G. Wang, M. A. Okoth, and L. C. Jiao, "Complex-valued convolutional autoencoder and spatial pixel-squares refinement for polarimetric SAR image classification," *Remote Sensing*, vol. 11, no. 5, 2019.
- [45] M. Chen, X. Shi, Y. Zhang, D. Wu, and M. Guizani, "Deep features learning for medical image analysis with convolutional autoencoder neural network," *IEEE Transactions on Big Data*, p. 1, Article ID 2717439, 2017.
- [46] K. Sugimoto, Y. Kon, S. Lee, and Y. Okada, "Detection and localization of myocardial infarction based on a convolutional autoencoder," *Knowledge-Based Systems*, vol. 178, pp. 123–131, 2019.
- [47] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.
- [48] J. Liu, X. Jin, Y. Hong et al., "Collaborative linear manifold learning for link prediction in heterogeneous networks," *Information Sciences*, vol. 511, pp. 297–308, 2020.
- [49] X. Liu, M. Wang, Z.-J. Zha, and R. Hong, "Cross-modality feature learning via convolutional autoencoder," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 15, no. 1, pp. 1–20, Article ID 3231740, 2019.
- [50] B. Leng, S. Guo, X. Zhang, and Z. Xiong, "3D object retrieval with stacked local convolutional autoencoder," *Signal Processing*, vol. 112, pp. 119–128, 2015.
- [51] G. Nikmehr, M. Salehi, and M. Jalili, "TSS: temporal similarity search measure for heterogeneous information networks," *Physica A: Statistical Mechanics and its Applications*, vol. 524, pp. 696–707, 2019.
- [52] S. Mohammadreza and G. Nikolas, "Dynamic clustering and propagation of congestion in heterogeneously congested urban traffic networks," *Transportation Research Part B: Methodological*, vol. 105, pp. 193–211, 2017.
- [53] Y. Li, Y. L. Liu, and K. Zou, "Research on the critical value of traffic congestion propagation based on coordination game," *Procedia Engineering*, vol. 137, pp. 754–761, 2016.
- [54] M. S. Seyfioglu, A. M. Ozbayoglu, and S. Z. Gurbuz, "Deep convolutional autoencoder for radar-based classification of similar aided and unaided human activities," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 4, pp. 1709–1723, 2018.
- [55] V. K. Ojha, A. Abraham, and V. Snášel, "Metaheuristic design of feed forward neural networks: a review of two decades of research," *Engineering Applications of Artificial Intelligence*, vol. 60, pp. 97–116, 2017.