Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 8867776, 12 pages
https://doi.org/10.1155/2021/8867776

Research Article

DCAST: A Spatiotemporal Model with DenseNet and GRU Based on Attention Mechanism

Liyan Xiong,[1] Lei Zhang ,[1] Xiaohui Huang,[1] Xiaofei Yang,[2] Weichun Huang,[1] Hui Zeng,[1] and Hong Tang[1]

[1]*School of Information Engineering, East China Jiaotong University, Nanchang 330013, China*
[2]*School of Faculty of Science and Technology, University of Macau, E11, Macau 999078, China*

Correspondence should be addressed to Lei Zhang; 18702624900@163.com

Received 14 September 2020; Revised 10 January 2021; Accepted 4 February 2021; Published 22 February 2021

Academic Editor: Gordon Huang

The accurate prediction of crowd flow in urban areas is becoming more and more important in many fields such as traffic management and public safety. However, the complex spatiotemporal relationship of the traffic data and the influence of events, weather, and other factors makes it very difficult to accurately predict the crowd flow. In this study, we propose a spatiotemporal prediction model that is based on densely connected convolutional networks and gated recurrent units (GRU) with the attention mechanism to predict the inflow and outflow of the crowds in regions within a specific area. The DCAST model divides the time axis into three parts: short-term dependence, period rule, and long-term dependence. For each part, we employ densely connected convolutional networks to extract spatial characteristics. Attention-based GRU module is used to capture the temporal features. And then, the outputs of the three parts are fused by weighting elementwise addition. At last, we combine the results of the fusion and external factors to predict the crowd flow in each region. The root mean square errors of the DCAST model in two real datasets of taxis in Beijing (TaxiBJ) and bikes in New York (BikeNYC) are 15.70 and 5.53, respectively. The experimental results show that the results are more accurate and reliable than that of the baseline model.

1. Introduction

Crowd flow prediction based on big data is a typical spatiotemporal prediction problem, which is of great significance of social stability [1–3]. In essence, crowd flow prediction is based on historical data to predict the relevant value of crowd flow in a region. If the crowd flow in a certain area can be predicted in advance, measures can be taken in advance to ensure safety of the crowds and reduce the loss caused by traffic congestion.

There are three difficulties in accurately predicting the flow of crowds in an area. (1) Spatial factor: crowds inflow and outflow in r_2 (as shown in Figure 1(a)) are not only affected by surrounding areas r_1 and r_3 but also affected by crowd flow in geographically distant areas. (2) Temporal factor: crowd flow in an area is usually periodic (as shown in Figure 1(b)). For example, crowd flow is relatively high during morning and evening rush hours and repeats roughly every 24 hours, and the morning rush hour starts later and later as the temperature drops. (3) External factors: weather and some events may have a significant impact on the flow of crowds in an area. For example, when a great party is held, people gather in large numbers in one area. The influence of various factors makes it difficult to predict the crowd flow.

This study mainly studies the crowd flow prediction problem with the region [4], taking into account two types of crowd inflow and outflow. As shown in Figure 1(a), inflow refers to the number of crowds entering a region in a certain time interval, while outflow refers to the number of crowds coming out of a region in a certain time interval. Both types reflect changes in crowd flow, so it is significant to control these two types. We can obtain crowd flow data onto vehicle track data, mobile phone signal data, public transportation data, and pedestrians.

Many researchers solve crowd flow prediction problems mostly based on mathematical equations or simulation

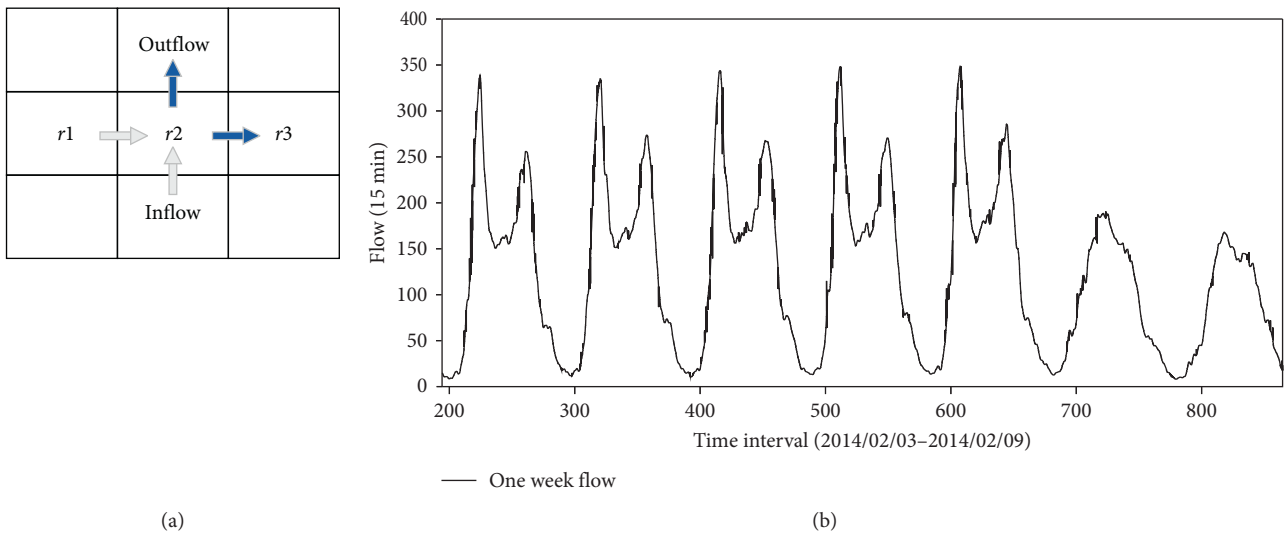

Figure 1: Crowd flow in an area. (a) Inflow and outflow. (b) Weekly change rule of crowd flow.

technology, while the real crowd flow involves the weather, events, crowds, and other factors, and it is difficult to accurately express with mathematical models. Traffic big data have become a basic resource with rich content and complex structure, and we must make good use of these resources. However, classical shallow learning algorithms cannot adapt well to the new situation. Williams et al. used ARIMA [5] to model and predict vehicle traffics to flow. Castro-Neto et al. [6] proposed the application of online support vector machine for regression supervised statistical learning technology to expressway short-term traffic flow prediction. Sun et al. [7] proposed a short-time traffic flow prediction method based on the Bayesian network, and the traffic flow between adjacent road links to the traffic network is modeled as the Bayesian network. Chen et al. [8] presents a novel social media-based approach to traffic congestion monitoring. However, these methods fail to capture the complex temporal and spatial correlations in the data. Therefore, the prediction of crowd flow needs to involve a data-driven model [9]. The most representative data-driven model is deep learning [10], which can automatically extract relevant depth characteristics of data. For example, Zhang et al. [4] proposed a crowd flow prediction method based on CNN to predict the crowd flow in urban areas after the grid. Yao et al. [11] proposed a spatiotemporal dynamic network, introduced a gate mechanism of traffic flow to learn the dynamic similarity between locations, and designed a periodic shift attention mechanism to deal with long-term periodic shifts. However, this model is larger and more complex.

To address the previously mentioned problem, we present a traffic prediction model DCAST based on deep learning. This model utilizes the DenseNets [12] module to capture spatial correlation. DenseNets connects each layer of CNN to each subsequent layer in a feed-forward manner, which can enhance feature reuse and enhance feature propagation. In terms of temporality, the attention-based mechanism-gated recurrent unit module is used to capture the potential time dependence on the DCAST data according to the temporal closeness to the predicted target. By combining the above modules, the model can not only better capture the temporal and spatial characteristics of data but also make full use of the external information. In this way, the crowd flow can be predicted more accurately, which is of great significance to improve the operation efficiency of the city and strengthen the management of urban public safety.

The model was validated against two real-world datasets (BikeNYC and TaxiBJ), including bike-sharing data onto New York City in 2014 and taxi data onto Beijing from 2013 to 2016. Compared with the existing methods, our model has better performance. The main contributions to this study are summarized as follows:

(1) We proposed a spatiotemporal prediction model that is based on a densely connected convolutional network and gated recurrent unit with the attention mechanism. The model is more robust and flexible when dealing with traffic forecasting.

(2) We designed a gated recurrent unit with the attention mechanism module to capture the temporal features.

(3) Verify the validity of the model through a large number of experiments on two real-world datasets. The proposed model has better predictive ability than the typical shallow learning model and other models based on deep learning.

The rest of the study is structured as follows: Section 2 introduces the related work. Section 3 defines the problem and describes the techniques used in this study. Section 4 makes a detailed analysis of the motivation and structural design of the proposed DCAST model. Section 5 presents the experimental results of the DCAST model in real traffic datasets and analyzes and evaluates the performance of the proposed method. The last section summarizes the whole study and looks forward to the future research direction.

2. Related Work

In recent years, spatiotemporal prediction has been widely concerned, and crowd flow prediction is a typical spatiotemporal prediction problem. This section discusses works related to spatiotemporal prediction issues.

The earliest work on this question focused on time series prediction. Williams et al. used ARIMA [5] to model and predict vehicle traffics to flow. Castro-Neto et al. [6] proposed the application of online support vector machine for regression supervised statistical learning technology to expressway short-term traffic flow prediction. Chan et al. [13] proposed an optimized ANN model to predict short-term traffic flow by using mixed exponential smoothing and Levenberg–Marquardt algorithm. Sun et al. [7] proposed a short-time traffic flow prediction method based on the Bayesian network, and the traffic flow between adjacent road links to the traffic network is modeled as the Bayesian network. Chen et al. [8] presented a novel social-media based approach to traffic congestion monitoring. Bai and Chen [14] proposed a deep architecture to predict the short-term traffic flow in an urban traffic network. However, these methods fail to capture the complex temporal and spatial correlations in the data.

After the breakthrough in the research of Hinton et al. [15], deep learning models have a superior performance in the fields of computer vision [16], speech recognition [17], and natural language processing [18], and the crowd flow prediction methods based on deep learning has attracted the attention of many researchers. In the first category, full connection layers are stacked, and data from multiple sources are combined. For example, Hua Wei et al. [19] proposed a zero-grid ensemble spatiotemporal model to predict traffic demand on four predictors, and Dong Wang et al. [20] presented an end-to-end framework called deep supply demand using a novel deep neural network structure to find the gap between taxi supply and demand. These methods use a large number of features but do not explicitly model the interspace and time interactions. In the second category, convolution structure is applied to capture the spatial correlation between space and time prediction. For instance, Zhang et al. [4] proposed a deep learning-based prediction model for spatiotemporal data (DeepST), in which spatiotemporal component employs the framework of convolutional neural networks to simultaneously model spatial near and distant dependencies and so on. Zhang et al. [21] then presented an end-to-end structure of ST-ResNet, which employs the residual neural network to model the temporal closeness, period, and trend properties of crowd traffic. Ziru Xu et al. [22] proposed a PredCNN model that was completely based on CNN, which used the multiplicative cascade unit to predict the future image without any recursive chain structure. These methods take into account the influence of spatial factors but do not fully consider the influence of temporal factors. In the third category, the model based on the recurrent neural network is used to model the sequential dependent relationship. For example, H. Yao et al. [11] proposed a spatial-temporal dynamic network, which uses a traffic gating mechanism to learn the dynamic similarity between locations and designed a periodic attention mechanism to deal with long-term periodic time transfers. Sønderby et al. [23] proposed splicing CNN and LSTM module for convolution LSTM dependence of space and time to deal with the taxi demand forecast. He et al. [24] proposed a spatiotemporal attentive neural network for the networkwide and long-term traffic prediction, which exploit a codec system structure with the attention mechanism to forecast traffic speed. Le Nguyen and Ji [25] used the convLSTM structure to solve the traffic matrix prediction problem and fully modeled the space-time model. Yao et al. [26] further proposed a multi-perspective space-time network for demand prediction, which integrates LSTM, local CNN, and structural embedding and comprehensively considers spatial, temporal, and semantic relations. Lin et al. [27] proposed a hybrid model called SpAE-LSTM, which uses a sparse autoencoder to extract the spatial characteristics of the spatial-temporal matrix through the full connected layers and cocaptures the spatial-temporal features of traffic flow evolution with the LSTM network for prediction. Li et al. [28] proposed a novel spatiotemporal prediction model that uses a densely connected convolutional network to extract spatial characteristics, a fully-connected network to extract features, and finally, an attention-based long short-term memory module is leveraged to capture the temporal pattern. Zhang et al. [29] proposed a novel spatial-temporal cross-domain neural network to effectively capture the complex patterns hidden in cellular data and adopted a convolutional long short-term memory network as its subcomponent, which has a strong spatiotemporal correlation modeling capability. Due to the large memory consumption and calculation amount of the LSTM module, Li et al. [30] extended the traditional CNN and RNN structures to the graph-based CNN and RNN for traffic prediction, such as the graph convolution GRU. In the above study, the spatial correlation between regions is based on one perspective, ignoring the different importance of each time interval. Du et al. [31] proposed a hybrid multimodal flow prediction model (HMDLF) based on deep learning, which combines CNN, GRU, and attention mechanism. However, the DCAST model exceeds the ability of HMDLF to capture temporal and spatial correlation.

The DCAST model automatically learns the dynamic temporal and spatial dependent features of crowd traffic data through the attention-based GRU module and the densely connected convolutional neural network.

3. Preliminaries

3.1. Problem Definition. Based on previous studies [21], we divide the whole city into a $a \times b$ grid map with n regions, where $n = a \times b$, and a grid represents a region. The spatiotemporal prediction problem is measured in a variety of ways, including air quality [32], weather, taxi orders, and bike rental/return. Here, we study the inflow and outflow of crowd flow. The crowd flow trajectory at t in a certain time interval is recorded as a set P. The inflow and outflow of the grid region at the time interval can be defined as

$$x_t^{\text{in},a,b} = \sum_{T_r \in P} \left|\left\{i > 1 \middle| g_{i-1} \notin (a,b), g_i \in (a,b)\right\}\right|,$$
$$x_t^{\text{out},a,b} = \sum_{T_r \in P} \left|\left\{i \geq 1 \middle| g_i \in (a,b), g_{i-1} \notin (a,b)\right\}\right|, \quad (1)$$

where g_i is a geographic coordinate, $g_i \in (a,b)$ denotes that the point g_i lies in region (a,b); $T_r{:}g_1 \longrightarrow g_2 \longrightarrow \cdots \longrightarrow g_l$ represents the trajectory of the moving object r; l represents the trajectory length; and $|\cdot|$ is the cardinality of a set.

If the grid map is regarded as an image of length a and width b, the inflow and outflow in time interval t can be represented as a two-channel image $X_t \in R^{2\times a\times b}$, where $(X_t)_{0,a,b} = x_t^{\text{in},a,b}$ and $(X_t)_{1,a,b} = x_t^{\text{out},a,b}$.

As shown in Figure 2, the bar on the right represents the relationship between crowd flow and color brightness. The horizontal and vertical axes are used to identify specific areas.

Using the above notations, the crowd flow prediction problem can be defined as

Definition 1 (Crowd flow prediction). Given the historical data of crowd flows $\{X_t | t = 1, 2, \ldots, n\}$, predict X_{n+1}.

3.2. Attention Mechanism. As one of the most influential ideas in the field of deep learning, the attention mechanism aims to overcome the problem of information loss caused by fixed intermediate vector length when the length of input sequence is relatively long. It was originally designed for the seq2seq model in natural language processing (NLP) and has been rapidly applied to other fields since then. The output of the attention mechanism can be written as

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \quad (2)$$

where $U = XW_U$, $U \in (Q,K,V)$, and X is the input, W_U is learnable matrix, d_k denotes the dimension of keys, and K^T is the transpose of the matrix K. $\text{softmax}(\cdot)$ is an activation function that is defined as $\text{softmax}(x_i) = (e^{x_i}/\sum_j e^{x_j})$, where x_i is the i^{th}dimension of the N-dimensional vector $x(i, j = 1, 2, \ldots, N)$.

3.3. Gate Recurrent Unit. The problem of gradient disappearance or gradient explosion is easy to occur in the calculation of back propagation when the layers of the simple recurrent neural network (RNN) are relatively deep. Therefore, RNN sometimes fails to capture long-term dependencies on sequences. The LSTM proposed by Hochreiter and Schmidhuber [33] is a variant based on RNN, which can capture the long-term dependence features of sequence data. GRU is a variant network based on LSTM, proposed by Cho et al. [34] (Figure 3). It combines the forget gate and input gate in LSTM into update gate, maintaining the effect of LSTM while making the structure simpler. Therefore, we use GRU for long-term dependency features learning.

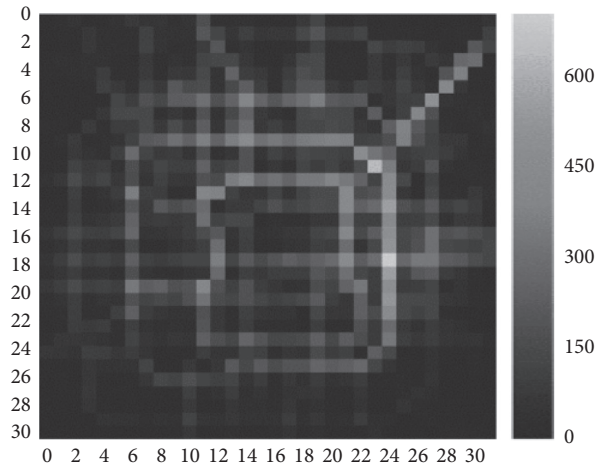

Figure 2: Heat map of crowd flow in Beijing.

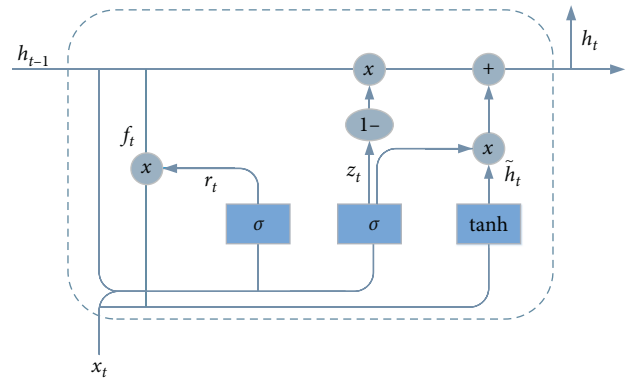

Figure 3: Typical GRU block diagram, visualized by http://colah.github.io/posts/2015-08-Understanding-LSTMs/.

Figure 3 is a typical GRU block diagram. GRU has only to update gate z_t and reset gate r_t. The long-term dependency learning block GRU calculates the hidden states through a set of equations formulated as

$$z_t = \sigma\left(W^{(z)} \cdot [h_{t-1}, x_t]\right),$$
$$r_t = \sigma\left(W^{(r)} \cdot [h_{t-1}, x_t]\right),$$
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right), \quad (3)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t.$$

In these equations, update gate z_t controls the extent to which states information from the previous moment is substituted into the current state, and reset gate r_t controls the extent to which state information from the previous moment is ignored. σ is the activation function. The candidate activation $\tilde{h}_t$ is computed with the reset gate r_t (which control how much of the previous information needs to be retained), and $*$ denotes the elementwise multiply operation. Finally, $\tilde{h}_t$ represents the actual activation of the proposed GRU unit at time$t$, which is a linear interpolation between the previous activation h_{t-1} and the candidate activation $\tilde{h}_t$.

4. DCAST Model

4.1. Overview of DCAST. As shown in Figure 4, we first processed the crowd flow of each region at the time interval t as an image of shape (2, a, b), and then, the time axis is divided into three parts, which are used to simulate the short-term dependence, period rule, and long-term dependence, respectively. The three parts share the same network structure with a densely connected convolutional network, and followed by the attention-based mechanism-gated recurrent unit module, get the different interval of timing the crowd flow characteristics of spatiotemporal. Features from external datasets such as weather conditions and events fed into a two-layer fully connected neural network. The outputs of the first three parts are fused, and the results of different parts are given different weights. The result of the fusion is integrated with the output of the external component and fed into an activation function to obtain the prediction.

4.2. Structure of the First Three Parts. The first three parts (i.e., short-term dependence, period rule, and long-term dependence) share the same network structure, which is composed of a densely connected convolutional network and an attention-based GRU module, as shown in Figure 5.

The city can be divided into many areas according to their different spatial positions. The crowd flow in nearby areas will influence each other. Including areas with weak correlation will hinder the prediction of the performance of the target area and will waste the central feature of CNN. In order to solve these problems, inspired by Huang [12], we use a DenseNet module to capture the spatial correlation between all regions. As shown in Figure 5(a), the DenseNet structure is implemented by transferring all outputs of the $i-1$ layer to i^{th} layers.

The short-term dependency part in Figure 4 simulates the short-term dependency with several 2-channel matrices of the recent time interval. We first connected the short-term dependent sequence with the first axis (i.e., time interval) as a tensor $X_s^{(0)} \in R^{2l_s \times P \times Q}$ feed into k convolution layers. The transformation at each K layer is defined as

$$X_s^k = \sigma\left(\left(\sum_{l=0}^{l=k-1} X_s^l\right) * W_s^{(k)} + b_s^{(k)}\right), \tag{4}$$

where * denotes the convolutional operation, and $\sigma()$ is an activation function. In this study, we use the rectifier function as the activation function, e.g., the $\sigma(x) = \max(0, x)$, and $W_s^{(k)}$ and $b_s^{(k)}$ are the two learnable parameters in the k^{th} layer to be trained. The $\sum(\cdot)$ operation is to concatenate the input tensors in the first dimension (concatenate the channel dimension of images).

After k densely connected convolutional layers, we reshape the output $X_s \in R^{2l_s \times a \times b}$ into the feature vector $J_s \in R^{2l_s ab}$. In order to reduce the feature dimension, a dense layer is used to generate the final spatial feature S_s which can be written as

$$S_s = \sigma(J_s * W_s + b_s), \tag{5}$$

where W_s and $b_s$are the two learnable parameters sets.

The introduction of the attention mechanism into the GRU networks is to simplify the selection of input at the previous layers and is critical to each subsequent step. Figure 5(b) shows an attention-based GRU module that takes the spatial characteristics of each time interval as input, where the length of the input sequence is equal to 5. Select L time intervals to predict the next crowd flow. Combining the attention mechanism and GRU introduced above, the spatial-temporal feature ST in L time periods intervals are expressed as follows:

$$\text{ST}_{s+1} = f(ST_s, S_s), \tag{6}$$

where f is the GRU network, and S_s denotes the final spatial feature. The final feature of short-term dependence on attention mechanism is defined as follows:

$$\text{Final}_s = \sum_{j-1}^{j=L} a_j h_j, \tag{7}$$

where weight a_j measures the importance of the time interval j$\in${1, 2, …, L}, and h_j is the hidden state in the GRU module at time interval j. The important measure weight a is obtained by learning the spatiotemporal characteristics and the previously hidden state.

By doing the same operations as above, we can construct the period rule and long-term dependent parts of Figure 4. Suppose the length of the period rule sequence is l_p, the period is p, so the period rule dependent sequence is $[X_{t-(l_p-1)\cdot p}, \ldots, X_{t-1} \cdot p, X_t]$ such as formulas (4)–(7). Through the densely connected convolutional network and an attention-based GRU module, the output of the period rule part is Final_p. Similarly, the long-term dependent sequence length is l_{lt} and the long-term span is lt, so the long-term dependent sequence is $[X_{t-(l_{\text{lt}}-1)\cdot \text{lt}}, \ldots, X_{t-1} \cdot \text{lt}, X_t]$, and the output of the long-term dependent part is Final_{lt}. Note that p and lt are two different types of period spans. In the concrete implementation, p stands for daily periodicity and lt stands for weekly periodicity.

4.3. External Features Part and Fusion. Crowd flow forecasts are influenced by external factors such as weather, holidays, and event information. This part is optional, depending on whether the original data contain external information. Let E_t be the feature vector that predicts the external factor at time interval t. The implementation takes into account weather, holiday events, and metadata (i.e., Sundays, weekdays, and weekends). Formally, we stack the two-layer fully connected neural network; the first layer can be seen as an embedded layer for each subfactor and then activated. The second layer is exploited to map low to high dimensions with the same shape as X_t. The output of the external part is denoted as X_{E_t} in Figure 4.

Next, we discuss how to fuse the four parts of Figure 4, first using the parameter matrix fusion method to fuse the first three parts (i.e., short-term dependence, period rule,

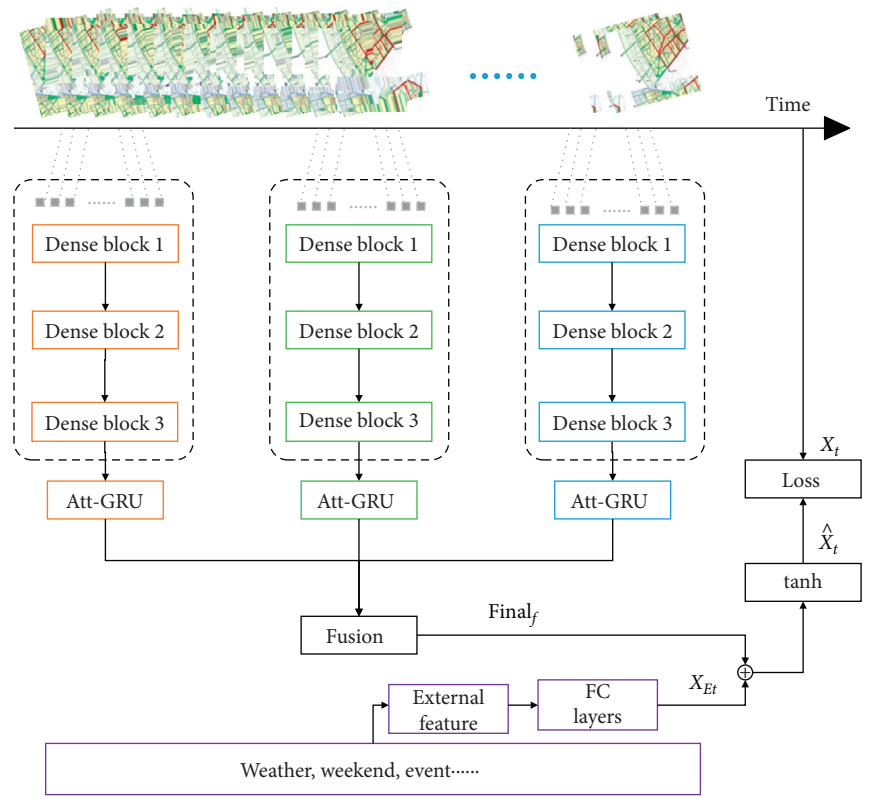

Figure 4: Structure diagram of the DCAST model.

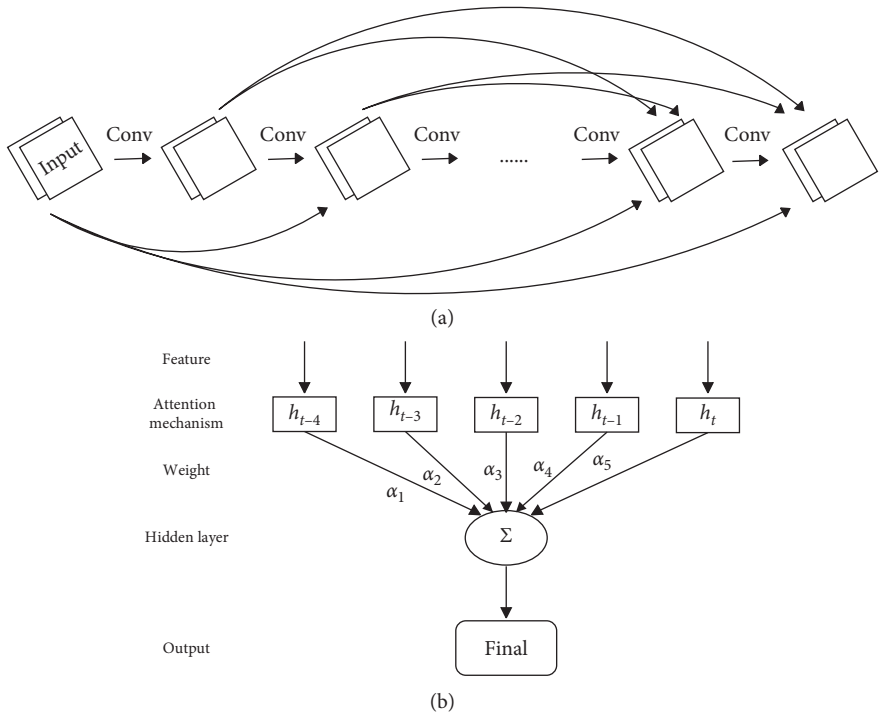

Figure 5: A dense block's internal configuration and an attention-based GRU module. (a) A dense block's internal configuration. (b) GRU based on the attention mechanism.

and long-term dependence) and then further combining them with the external part. The fusion Final_f of the first three parts is expressed as

$$\text{Final}_f = W_s \circ \text{Final}_s + W_p \circ \text{Final}_p + W_{\text{lt}} \circ \text{Final}_{\text{lt}}, \quad (8)$$

where $\circ$ is the Hadamard product (i.e., elementwise multiplication), and W_s, W_p, and W_{lt} are the learnable parameters that adjust the short-term dependence, period rule, and long-term dependence, respectively.

As shown in Figure 4, the fusion Final_f of the first three parts is integrated with external part $\widehat{X}_t$ as follows:

$$\widehat{X}_t = \tanh\left(\text{Final}_f + X_{E_t}\right), \quad (9)$$

where tanh is a hyperbolic tangent function, ensuring that the output value is [−1–1].

The DCAST model minimizes the mean square error between the predicted value and the real value through training:

$$\ell(\theta) = \left\| X_t - \widehat{X}_t \right\|_2^2, \quad (10)$$

where θ is the learnable parameter of the DCAST model.

4.4. Model Training (Algorithm). Algorithm 1 summarizes the training process of the DCAST model. We build a sample set D_{sample} (lines 2–7) from historical observations and then divide D_{sample} into the training set D_{train} and the testing set D_{test}. The former is used for the training model and the latter for the testing model. A batch of training samples D_{batch} is selected at each iteration to optimize the objective function (formula (10)) (lines 10–13).

5. Experiment and Results

5.1. Dataset Description. In this study, we use two large real datasets from New York City and Beijing to evaluate the proposed model. The details of each dataset are as follows:

BikeNYC: the bicycle trajectory data are taken from New York City [26] in 2014 from April 1 to September 30 (183 days). No external information is provided, but datasets on inflow and outflow and their respective timings are included. The city was divided into a 16×8 grid map, and the data of the last 10 days were selected as the test data and the data of the other days as the training data.

TaxiBJ: the taxis trajectory data are taken from Beijing in four time intervals: July 1, 2013–October 30, 2013, March 1, 2014–June 30, 2014, March 1, 2015–June 30, 2015, and November 1, 2015–April 10, 2016. The external information contains data about holidays and weather conditions. The experiment divided the city into 32×32 grid map and selected the data of the last four weeks as the test data and the data of other times as the training data.

5.2. Evaluation Criteria. We choose the mean square error (RMSE) and the mean absolute error (MAE) to evaluate the experimental results, which is defined as

$$\text{RMSE} = \sqrt{\frac{1}{M}\sum_{i=1}^{M}\left(v_i - \widehat{v}_i\right)}, \quad (11)$$

where v_i and $\widehat{v}_i$ are the ground truth and the predicted value, and M is the number of all predicted values.

$$\text{MAE} = \frac{1}{M}\sum_{i-1}^{M}\left|\left(v_i - \widehat{v}_i\right)\right|, \quad (12)$$

where v_i and $\widehat{v}_i$ are the ground truth and the predicted value, and M is the number of all predicted values.

5.3. Baseline Models. In this experiment, we compare the DCAST model with four traditional models and three based on the deep learning model:

(i) HA: the historical average method (HA) through the period before the crowd inflow and outflow of average to predict population flow. For example, for predicting crowd flow in an area from 7:30 pm to 8:00 pm on Friday, we can use all the actual data from that area from 7:30 pm to 8:00 pm on Friday. The HA model is simple and easy to manage but imprecise.

(ii) ARIMA: the autoregressive integrated moving average model (ARIMA), which is composed of autoregressive and moving average models, is the most widely used typical model in the field of time series prediction.

(iii) SARIMA: the seasonal autoregressive integrated moving average model combines the seasonal difference with the ARIMA model for modeling time series data with periodic characteristics.

(iv) VAR: vector autoregression (VAR) is usually used to estimate the dynamic relationship between the joint endogenous variables. It can predict the spatial and temporal data with many parameters and a large amount of calculation.

(v) ST-ANN: the ST-ANN extract spatial feature (place the area to be predicted in the center of the 3×3 unit) and the temporal feature (the previous time period) and input them into the artificial neural network. In this experiment, the previous time period was set as 8.

(vi) DeepST: a model for end-to-end prediction of spatiotemporal data. There are four variants, including DeepST-C, DeepST-CP, DeepST-CPT, and DeepST-CPTM, which focus on different time dependence and external factors, respectively.

(vii) ST-ResNet: the ST-ResNet model use convolutional neural networks and residual networks to predict the crowd flow in the whole city.

(viii)PredCNN: PredCNN is an entirely CNN-based architecture that models the dependencies between the next frame and the sequential video inputs. The cascade multiplicative unit (CMU) in PredCNN provides relatively more operations for previous video frames. And the CMU enables PredCNN to predict future spatiotemporal data without any recurrent chain structures.

The DCAST model is compared with other baseline models according to whether the model considers the spatial, temporal, and external characteristics of the data. As shown in Table 1, the DCAST model considers the characteristics of all aspects of the data.

5.4. Preprocessing and Parameters. In the output of the model DCAST, tanh is selected as the final activation, which ranges from −1 to 1. And, we use the min-max normalization to scale the crowd flow to the range [−1, 1].

In the evaluation, we scaled the predicted value back to the normal value and compared it with the ground truth. For external information, one-hot encoding is used to transform discrete features (i.e., weather and holidays), and min-max normalization is used to scale the data to the range of [0, 1]. The linear transformation of initial data is as follows:

$$x_i^* = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}}, \tag{13}$$

where x_i represents the sample from initial data; $x_{\max}$ and $x_{\min}$ represent the maximum and minimum values in the data, respectively; x_i^* denotes the transformed data. The predicted values of the model are rescaled back to produce the true predicted values.

This experiment runs on 1080Ti GPUs and uses Python 3.6 environments with TensorFlow and Keras (https://github.com/fchollet/keras) to build this model. As shown in Table 2, the DenseNets part contains 3 dense blocks and 32 filters of size (3, 3) in each dense block. The attention-based GRU part has two layers of GRU, and the number of neurons in each hidden layer is 128. In order to maintain the generalization ability of the DCAST model, we add a dropout layer that dropout rate is 0.5 after the dense layer. The batch size is set as 512, and the maximum epoch is set as 150. We employ the Adam [35] optimization model with a learning rate of 0.001. This optimizer had good universality and rapid convergence ability in the deep learning model. The decay of the learning rate is 0.001. We use 90% of the data as the training set and the rest 10% as the validation set in the training process. The early stop method with a patience of 20 is used to get the best results and avoid overfitting.

5.5. Performance Comparison with the Baseline Models. Table 3 provides the comparison among the DCAST model proposed in this study with other baseline models. The DCAST model reached the lowest RMSE of these models, BikeNYC (5.53) and TaxiBJ (15.70), respectively. Compared with the baseline models, RMSE of the DCAST model decreased BikeNYC (6.9%–52.98%) and TaxiBJ (1.26%–72.79%), respectively. It can be seen from the table that the HA model has the worst forecast results, with RMSE being BikeNYC (11. 76) and TaxiBJ (57. 69) because it only relies on historical data without considering spatial correlation and external information. Models based on deep learning perform better than traditional methods because they make more efficient use of data and information. As shown in Table 3, the prediction results of ST-ResNet and DeepST are superior to the above models as ST-ResNet and DeepST use CNNs to capture spatial information and consider time temporal periodicity. There are no any recurrent chain structures of PredCNN, which can predict future spatio-temporal data and achieve full parallelization, but it also loses some historical information.

In this study, the DCAST model of capturing the spatial-temporal characteristics between regions using a densely connected convolutional network, and an attention-based GRU outperforms the previously mentioned methods.

Figure 6 shows the variation diagram of loss in the DCAST model. It can be seen that when the epoch approaches 150, the value of loss does not change much. So, the early stop method with a patience of 20 is used to get the best results and avoid overfitting.

5.6. Performance Comparison with Model Variants. We study the influence of different parts in the DCAST model to confirm their validity. As the BikeNYC dataset does not contain external auxiliary information, we use the TaxiBJ dataset to carry out experiments and verify each part by deleting or replacing to form relatively complete comparison results.

As shown in Table 4, the model of CNN has the worst prediction effect, which only focuses on the spatial correlation of nearby areas. After replacing CNN with DenseNets structure, the prediction effect is increased by 20. 3%, proving that DenseNets can simulate complex spatial relations better than CNN. In addition, adding the GRU module or LSTM module to DenseNets improves the performance of DenseNets and verifies the validity of the temporal information.

DenseNets with attention GRU or LSTM outperforms the model variant without the attentional mechanism, proving that the attention mechanism can help GRU or LSTM better capture time patterns. As can be seen from Table 4, using GRU as the basic module is better than using LSTM as the basic module because GRU is a variant based on LSTM, which is simpler in structure and has fewer parameters while retaining a good effect. The DCAST model proposed in this study combines the densely connected convolutional network, GRU, attention mechanism, and external features to get the best prediction results. Therefore, the external feature from auxiliary information is helpful for prediction.

In addition, we also study the variations of RMSE of different epochs on different models. As shown in Figure 7,

Input: historical observations: $\{X_1, X_2, \ldots, X_{n-1}, X_n\}$; length of short-term dependence, period rule, and long-term dependence: l_s, l_p, l_{lt}; span of period rule and long-term dependence: p, lt; external features: $[E_1, E_2, \ldots, E_{n-1}, E_n]$
Output: DCAST, model
//Generate samples from historical crowd flow observations
(1) $D_{\text{sample}} \leftarrow \phi$
(2) For all available time interval t do
(3) $P_s = [X_{t-(l_s-1)}, \ldots, X_{t-1}, X_t]$
(4) $P_p = [X_{t-(l_p-1)\cdot p}, \ldots, X_{t-1\cdot p}, X_t]$
(5) $P_{\text{lt}} = [X_{t-(l_{\text{lt}}-1)\cdot \text{lt}}, \ldots, X_{t-1\cdot \text{lt}}, X_t]$
(6) Put a training instance $(\{P_s, P_p, P_{\text{lt}}, E_t\}, X_t)$ into D_{sample}
(7) End
(8) Divide D_{sample}into D_{train} and D_{test}
//Train the model
(9) Initialize all the parameters θ in DCAST
(10) Repeat
(11) Randomly choose a batch of samples D_{batch} from D_{train}
(12) Find θ by minimizing the objective (10) with D_{batch}
(13) Until stopping criteria is met
(14) Output the learned DCAST model

Algorithm 1: DCAST algorithm training process.

Table 1: Comparison of the DCAST model and baseline models.

Model	Temporal	Spatial	External
HA	Y	N	N
ARIMA	Y	N	N
SARIMA	Y	N	N
VAR	Y	N	N
ST-ANN	Y	Y	Y
DeepST	Y	Y	Y
ST-ResNet	Y	Y	Y
PredCNN	Y	Y	Y
DCAST	Y	Y	Y

Table 2: Parameter settings of the DCAST model.

Parameters	Value
Number of dense blocks	3
Number of GRU layers	2
Number of filters	32
Filters size	(3, 3)
Number of GRU hidden layer neurons	128, 128
Dropout rate	0.5
Learning rate	0.001
Batch size	512
Epoch	150
Optimizer	Adam

the RMSE variation of the DCAST model for different epochs is compared with other model variants. With the increase of the epochs, the predictive effect of all models improved, while the DCAST model remained superior to other model variants. We observe that the RMSE remained almost stable when the epoch is greater than 150 but did not change much when the epoch continued to grow. In other words, more epochs do not mean better prediction result; the generalization ability is not significantly improved when

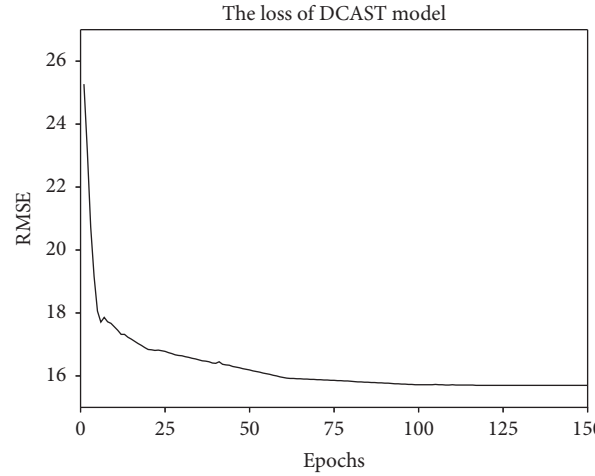

Figure 6: The loss of the DCAST model.

the epoch is greater than 150, and all models seem a bit overfitting when the epoch is greater than 250. To sum up, although the increase of the epoch can improve the precision of model training, it will lead to the problem of overfitting,

Table 3: BikeNYC and TaxiBJ results compared with the baseline models (in order to speed up the experiment, the results of reference [21] provide some baseline models).

Dataset	Model	RMSE	MAE
BikeNYC	HA	11.76	–
	ARIMA	10.07	–
	SARIMA	10.56	–
	VAR	9.92	–
	DeepST	7.43	–
	ST-ResNet	6.33	2.95
	PredCNN	5.94	2.82
	DCAST	5.53 ± 0.10	2.64
TaxiBJ	HA	57.69	–
	ARIMA	22.78	–
	SARIMA	26.88	–
	VAR	22.88	–
	ST-ANN	19.57	–
	DeepST	18.18	–
	ST-ResNet	16.69	9.71
	PredCNN	15.90	9.22
	DCAST	15.70 ± 0.10	9.13

Table 4: Comparison with model variants in TaxiBJ.

Model	RMSE
CNN	22.84
DenseNets	18.21
DenseNets + LSTM	17.27
DenseNets + GRU	16.89
DenseNet + LSTM + attention	16.33
DenseNet + GRU + attention	16.01
DCAST	15.70

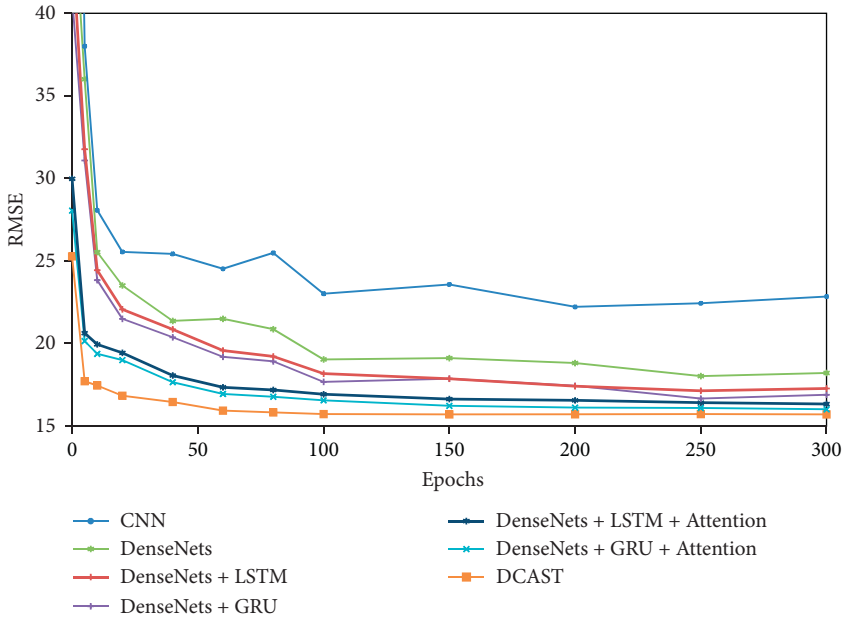

Figure 7: RMSE of the proposed DCAST model versus different epochs and comparisons with other models.

but the calculation is heavy, which is not conducive to the application of the model.

6. Conclusions and Future Research

In this study, a new spatiotemporal prediction model based on densely connected convolutional networks and gated recurrent units with attention is proposed for crowd flow prediction. The DCAST model divides the time axis into three parts: short-term dependence, period rule, and long-term dependence. For each part, based on historical data, weather, and events, we use a dense convolutional network to capture spatial dependency and design a GRU based on attention mechanism that captures temporal dependency. The fusion results of these three parts are further combined with the external features extracted from the external auxiliary information. We innovatively combined these deep learning techniques to build a new traffic forecasting model that is more robust and flexible. The model can extract complex spatiotemporal features hidden in depth. Two types of population flow in Beijing and New York were evaluated. The experimental results showed that the prediction results of the DCAST model were significantly better than those of the 7 benchmark models, which proved that the model was more suitable for the prediction of population flow.

In the future studies, we plan to use the graphic neural network (GNN) [36, 37] to further study the dynamic correlation between regions. In real life, there are no rules for the division of areas. Therefore, the GNN is more effective than CNN in capturing complex space-temporal characteristics and obtaining stable prediction results. In addition, other types of data will be considered in the future, and an appropriate fusion mechanism will be used to better fuse different types of data, so as to achieve accurate prediction of regional crowd flow.

Data Availability

The research data used to support the findings of this study are from the study "Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction," from AAAI 2017 https://arxiv.org/pdf/1610.00081.pdf, and from https://www.jianguoyun.com/p/DesHv2UQs-HRBxi5gtYB.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (61967006, 62062033, and 62067002), the Education Department project of Jiangxi Province (GJJ190317), and the Natural Science Foundation of Jiangxi Province (20192ACBL21006).

References

[1] G. Shen, C. Chen, Q. Pan, S. Shen, and Z. Liu, "Research on traffic speed prediction by temporal clustering analysis and convolutional neural network with deformable kernels (may, 2018)," *IEEE Access*, vol. 6, no. 2169-3536, pp. 51756–51765, 2018.

[2] M. Zheng, T. Li, R. Zhu et al., "Traffic accident's severity prediction: a deep-learning approach-based CNN network," *IEEE Access*, vol. 7, no. 2169-3536, pp. 39897–39910, 2019.

[3] M. X. Hoang, Y. Zheng, and A. K. Singh, "Forecasting citywide crowd flows based on big data," in *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS'16*, California, CA, USA, November 2016.

[4] J. Zhang, Y. Zheng, D. Qi et al., "DNN-based prediction model for spatio-temporal data," in *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS'16*, California, CA, USA, November 2016.

[5] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: theoretical basis and empirical results," *Journal of Transportation Engineering*, vol. 129, no. 6, pp. 664–672, 2003.

[6] M. Castro-Neto, Y.-S. Jeong, M.-K. Jeong, and L. D. Han, "Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions," *Expert Systems with Applications*, vol. 36, no. 3, pp. 6164–6173, 2009.

[7] S. Sun, C. Zhang, and G. Yu, "A bayesian network approach to traffic flow forecasting," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 1, pp. 124–132, 2006.

[8] P.-T. Chen, F. Chen, Z. Qian, and C. Shenzhen, "Road traffic congestion monitoring in social media with hinge-loss Markov random fields," 2014.

[9] J. Zhang, F.-Y. Wang, K. Wang, W.-H. Lin, X. Xu, and C. Chen, "Data-driven intelligent transportation systems: a survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1624–1639, 2011.

[10] J. Schmidhuber, "Deep learning in neural networks: an overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[11] H. Yao, X. Tang, H. Wei et al., "Revisiting spatial-temporal similarity: a deep learning framework for traffic prediction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 5668–5675, New York, NY, USA, February 2019.

[12] G. Huang, Z. Liu, L. Van Der Maaten et al., "Densely connected convolutional networks," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 2261–2269, Honolulu, HI, USA, July 2017.

[13] K. Y. Chan, T. S. Dillon, J. Singh, and E. Chang, "Neural-network-based models for short-term traffic flow forecasting using a hybrid exponential smoothing and levenberg-marquardt algorithm," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 2, pp. 644–654, 2012.

[14] J. Bai and Y. Chen, "A deep neural network based on classification of traffic volume for short-term forecasting," *Mathematical Problems in Engineering*, vol. 2019, Article ID 6318094, 10 pages, 2019.

[15] 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.

[16] A. Krizhevsky, I. Sutskever, and G. Hinton, *Imagenet Classification With Deep Convolutional Neural Networks*, NIPS Curran Associates Inc, Red Hook; NY;USA, 2012.

[17] A. Graves, A. R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proceedings of the Acoustics Speech & Signal Processing. Icassp. International*

Conference, pp. 6645–6649, Vancouver, BC, Canada, June 2013.

[18] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," *31st International Conference on Machine Learning*, vol. 4, pp. 2931–2939, 2014.

[19] H. Wei, Y. Wang, T. Wo et al., "Zest: a hybrid model on predicting passenger demand for chauffeured car service," in *Proceedings of the Conference on Information and Knowledge Management*, pp. 2203–2208, Indianapolis, IN, USA, October 2016.

[20] D. Wang, W. Cao, J. Li et al., "DeepSD: supply-demand prediction for online car-hailing services using deep neural networks," in *Proceedings of the 2017 IEEE 33rd International Conference on Data Engineering (ICDE)*, pp. 243–254, San Diego, CA, USA, June 2017.

[21] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pp. 1655–1661, San Diego, CA, USA, June 2017.

[22] Z. Xu, Y. Wang, M. Long, J. Wang, and M. KLiss, *PredCNN: Predictive Learning with Cascade Convolutions*, Stockholm, Sweden, 2018.

[23] S. K. Sønderby, C. K. Sønderby, H. Nielsen, and O. Winther, *Convolutional LSTM Networks for Subcellular Localization of Proteins, in Algorithms for Computational Biology*, ACM, Tarragona, Spain, 2015.

[24] Z. He, C.-Y. Chow, and J.-D. J. I. A. Zhang, "A spatio–temporal attentive neural network for traffic prediction," in *Proceedings of the 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, Washington, DC, USA, April 2019.

[25] P. Le Nguyen and Y. Ji, "Deep convolutional lstm network-based traffic matrix prediction with partial information," in *Proceedings of the 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, Washington, DC, USA, April 2019.

[26] H. Yao, F. Wu, J. Ke et al., "Deep multi-view spatial-temporal network for taxi demand prediction," in *Proceedings of the AAAI Conference on Artificial Intelligence, Hilton New Orleans Riverside*, pp. 2588–2595, New Orleans, Louisiana, USA, February 2018.

[27] F. Lin, Y. Xu, Y. Yang, and H. Ma, "A spatial-temporal hybrid model for short-term traffic prediction," *Mathematical Problems in Engineering*, vol. 2019, Article ID 4858546, 12 pages, 2019.

[28] W. Li, W. Tao, J. Qiu, X. Liu, X. Zhou, and Z. Pan, "Densely connected convolutional networks with attention LSTM for crowd flows prediction," *IEEE Access*, vol. 7, pp. 140488–140498, 2019.

[29] C. Zhang, H. Zhang, J. Qiao et al., "Deep transfer learning for intelligent cellular traffic prediction based on cross-domain big data," *IEEE Journal on Selected Areas in Communications*, vol. 37, pp. 1387–1401, 2019.

[30] Y. Li, R. Yu, C. Shahabi et al., "Diffusion convolutional recurrent neural network: data-driven traffic forecasting," in *Proceedings of the International Conference on Learning Representations*, Vancouver Convention Center, Vancouver, BC, Canada, April 2018.

[31] S. Du, T. Li, X. Gong et al., "A hybrid method for traffic flow forecasting using multimodal deep learning," *International Journal of Computational Intelligence Systems*, vol. 13, no. 29, pp. 85–97, 2020.

[32] X. Yi, J. Zhang, Z. Wang et al., "Deep distributed fusion network for air quality prediction," in *Proceedings of the The 24th ACM SIGKDD International Conference*, New York, NY, USA, July 2018.

[33] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[34] K. Cho, B. Van Merriënboer, D. Bahdanau et al., "On the properties of neural machine translation: encoder-decoder approaches," in *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pp. 103–111, San Diego, CA, USA, October 2014.

[35] D. P. Kingma, J. Ba, and Adam, "A Method for Stochastic Optimization," in *Proceedings of Conference Paper at the 3rd International Conference for Learning Representations*, San Diego, CA, USA, 2015.

[36] P. W. Battaglia, J. B. Hamrick, V. Bapst et al., "Relational inductive biases, deep learning, and graph networks," 2018, https://arxiv.org/abs/1806.01261.

[37] J. Zhou, G. Cui, Z. Zhang et al., "Graph neural networks: a review of methods and applications," 2018, https://arxiv.org/abs/1812.08434.