Knowledge-Based Systems 250 (2022) 109054

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Deep spatio-temporal 3D densenet with multiscale ConvLSTM-Resnet network for citywide traffic flow forecasting

Rui He [a], Yanbing Liu [a,b,*], Yunpeng Xiao [a], Xingyu Lu [a], Song Zhang [a]

[a] College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing, 400065, China
[b] Chongqing Medical University, Chongqing, 400016, China

ARTICLE INFO

Article history:
Received 5 January 2022
Received in revised form 8 May 2022
Accepted 13 May 2022
Available online 24 May 2022

Keywords:
3D densenet
Traffic prediction
Spatio-temporal data mining
Neural network

ABSTRACT

Reliable traffic flow forecasting is paramount in Intelligent Transportation Systems (ITS) as it can effectively improve traffic efficiency and social security. Its vital challenge is to effectively integrate various factors (such as multiple temporal correlations, complex spatial correlation, high heterogeneous) to infer the evolution trend of future traffic flow. Inspired by spatio-temporal prediction in computer vision, we regard traffic data slices at each moment as "traffic frames". This paper presents an end-to-end architecture named Spatio-Temporal 3D Densenet Multiscale ConvLSTM-Resnet Network (ST-3DDMCRN) to predict future traffic flow accurately. Specifically, a 3D densenet network is applied simultaneously to capture the traffic frame's local regional spatio-temporal information. Traditional Resnet networks cannot capture long-range spatial correlation, a novel multiscale ConvLSTM-Resnet network is developed to overcome this problem, extracting traffic frame's nearby and long-range spatial dependencies. In addition, considering the spatio-temporal heterogeneity of traffic frames, a Region-Squeeze-and-Excitation (RSE) unit is designed to accurately quantify the difference of the contributions of the correlations in space. The experiment result on two datasets in the real world illustrates the ST-3DDMCRN model outperforms the state-of-art baselines for the citywide traffic flow prediction. Furthermore, to validate the model's generality, we utilize the model to predict the passenger pickup/dropoff demand task, the prediction results are more accurate than the baseline methods.

1. Introduction

With the development of social urbanization, traffic flow prediction, as an indispensable part of the ITS [1], is becoming increasingly important for traffic management, travel planning, and social security [2,3]. For instance, Brits wasted 115 h in traffic congestion in 2019, which cost £6.9 billion according to the traffic condition report released by Inrix, a transportation analytics company. A dangerous stampede[1] at a religious festival in Israel resulted in 45 deaths in April 2021. Metropolises such as New York and Beijing suffer from traffic issues related to time waste, air pollution, traffic congestion, and even traffic accidents every day. If we could precisely predict the future traffic flow throughout city areas, similar traffic jams or hot events could be prevented by giving early warnings in advance. ITS has been studied for decades to meet the above challenge and has effectively improved urban traffic efficiency. Recently, due to its great potential in many practical applications (such as intelligent traffic management, smart logistics, travel route planning), traffic flow prediction [4–6] has attracted widespread research interest in academia and industry.

This article aims to predict future citywide traffic flow with historical mobility data of bicycles and taxis. With the advancement of communication technology during the last few years, various vehicles (bikes, taxis, buses, etc.) have been equipped with GPS devices, which obtain considerable actual traffic trajectory data. These traffic trajectory data has both temporal and spatial properties. Therefore, how to use the massive spatio-temporal traffic data to predict the future traffic flow is still an open problem. It is necessary to fully consider the spatio-temporal correlation of traffic data and establish a robust traffic prediction model to predict the future traffic conditions. Recently, many scholars have started to investigate traffic data using data-driven methods. The data-driven strategies include traditional machine learning-based methods and deep learning-based methods. Due to the need for careful feature engineering, traditional machine learning methods may not capture high-dimensional features of traffic flow.

Inspired by different disciplines, deep learning-based methods are classified into two categories: one is a neural network

* Corresponding author at: Chongqing Medical University, Chongqing, 400016, China.
E-mail addresses: d200201008@stu.cqupt.edu.cn (R. He), liuyb@cqupt.edu.cn (Y. Liu).

[1] https://www.israelnationalnews.com/news/305300

https://doi.org/10.1016/j.knosys.2022.109054
0950-7051/

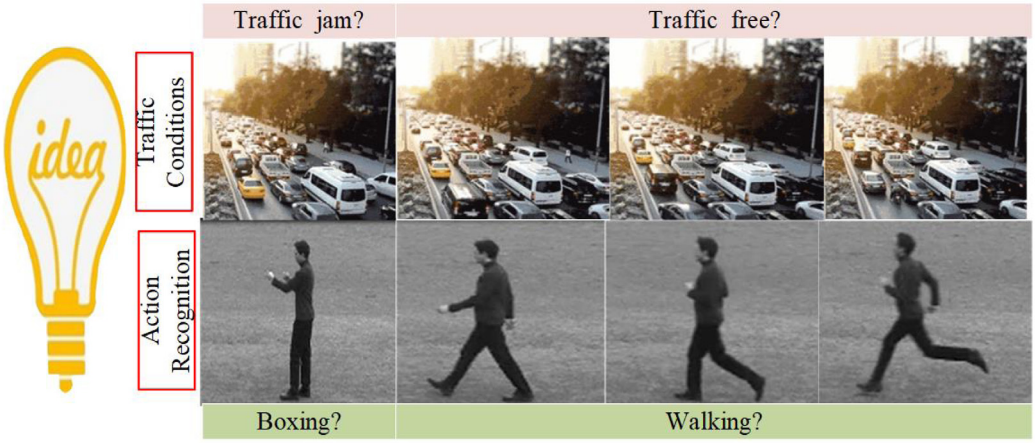


Fig. 1. The inspiration of this paper.

based on mathematical principles, and the other is a neural network based on biological principles. The former is primarily represented by the radial basis (RBF) neural network, which is primarily used to approximate nonlinear functions and deal with difficult-to-analyse regularities in the system [7–9]. The latter is principally represented by Convolutional Neural Networks (CNNs) that can obtain spatial structure information automatically and hierarchically via convolution operations. Therefore, many researchers use the two-dimensional convolutional neural network (2DCNN) technique to learn about the high-dimensional and nonlinear properties of traffic data [4–6,10].

However, because these 2DCNN-based methods ignore the correlation of continuous-time intervals so that the spatio-temporal information in the low layer may not be thoroughly explored. In recent years, some academics have begun to utilize 3D convolution to extract the spatio-temporal information simultaneously [11–13]. Nevertheless, the new problem is that the 3D convolution may not catch the whole length of the traffic data at one time which causes prediction accuracy to decrease.

Motivated by the success of the 3DCNN-ConvLSTM framework in spatio-temporal prediction in computer vision [14–16], the movement of vehicles has a similar spatio-temporal correlation to action recognition in Computer Vision. How to estimate a traffic jam or free, or how to judge whether the person's behaviour is boxing, or walking, as shown in Fig. 1. Nevertheless, we cannot directly apply this framework to predict traffic flow as it has multiple dynamic temporal dependencies, complex spatial dependencies, and spatio-temporal heterogeneity, which are essential in modelling. Therefore, how to predict future traffic flow has the following challenges:

- Multiple dynamic temporal dependencies. It is observed that the traffic flow in a region is affected in different dynamic time intervals (i.e., closeness, daily-period, weekly-period). That is, traffic frames have apparent daily and weekly periodicity. For instance, the rush hour generally lasts from 7 a.m. to 9 a.m. The peak time pattern is repetitive and easy to detect in big cities. Consequently, how to capture complex temporal dependencies is an issue.
- Complex spatial correlation. The complex spatial correlation includes local regional and long-range spatial dependencies in the traffic flow. On the one hand, residential and industrial regions may cause large changes when people from living areas commute to industrial areas for work. On the other hand, people can go anywhere quickly by BRT or subway in modern cities. It is challenging to design a network to capture local regional and long-range spatial correlation in modelling.
- High heterogeneity. Traffic flow exhibits heterogeneity in time and space. It shows the difference in spatial correlation contribution. Due to different periods and geographical locations, traffic flow presents similar cycles and different trend features.

To address the challenges mentioned above, the **S**patio-**T**emporal **3D** **D**ensenet **M**ulti-scale **C**onvlstm-**R**esnet **N**etwork (ST-3DDMCRN) is proposed for forecasting future citywide traffic flow accurately. The main innovations and contributions to our study are summarized as follows:

- The 3D densenet technology is first introduced into the traffic field to efficiently capture low-level spatio-temporal correlation features and dependency features among local regions. To the best of our knowledge, we exploit the 3D densenet network in traffic prediction problems for the first time.
- A multiscale ConvLSTM-Resnet network is proposed, effectively capturing local regional and long-range spatial dependencies through different dilated rates. This framework obtains multiscale receptive fields and better simulates spatio-temporal correlation.
- To capture multiple temporal patterns of traffic frames, the proposed ST-3DDMCRN model includes three modules, i.e., closeness, daily-period, and weekly-period. At the same time, a novel "RSE" unit is designed in each module that can automatically quantify different regions' contributions to better capture the traffic frames' heterogeneity.
- We prove the superiority of the ST-3DDMCRN model in traffic prediction on two popular datasets (BikeNYC and TaxiBJ). In addition, to show the model's generalization, we exploit the model to forecast future passenger pick up/off demand and achieve superior performance.

The rest of this paper is organized as follows. Section 2 presents a review of the work related to traffic forecasting. The problem of traffic flow prediction is formulated in Section 3. We provide a detailed analysis of the proposed model in Section 4. Section 5 contributes to the study of the experimental and comparison results. Section 6 concludes the work along with directions for future research.

2. Related work

2.1. Traffic prediction

In the past several decades, many scholars have begun studying traffic prediction due to the growingly severe traffic congestion problems in the metropolis. This section will review the research results achieved by scholars in recent years.

For temporal correlation of traffic flow, classical statistical methods such as Historical average (HA), Support Vector Machines (SVM),

Auto-Regressive Integrated Moving Average (ARIMA) [17] were exploited to solve traffic prediction problems. Subsequently, some academics utilized classical Recurrent Neural Networks (RNNs) [18] to capture the temporal dependencies. Liu et al. [19] exploited Long Short Term Memory (LSTM) networks to study short-term traffic prediction problems. Tian et al. [20] proposed a multimodal spatial–temporal GCN (graph convolution network) that utilized a gated cycle unit (GRU) to extract temporal features. However, these models primarily concentrate on temporal correlation and may not capture well complex spatial correlation.

For complex spatial correlation of traffic flow, many scholars utilized single-layer or multiple-layer convolution to simulate the spatial dependencies of traffic flow in most literature. Zhang et al. [4] introduced a spatio-temporal model (DeepST) to capture spatial dependencies by stacking multiple convolutions. Subsequently, Zhang et al. [5] developed a classic Residual Neural Network [21] to capture the spatial correlation with different temporal features (closeness, period, and trend). Lin et al. [22] employed the 2DCNN and ResPlus unit to simulate long-distance spatial correlation for crowd prediction. Nevertheless, these approaches fail to model multiscale spatial dependencies and complex nonlinear spatio-temporal relationships.

Recently, some academics have utilized hybrid methods, which combine 2D CNN with RNN or LSTM, to capture spatio-temporal features. Yao et al. [6] developed a novel Spatial–Temporal Dynamic Network (STDN) that combines 2DCNN, LSTM and periodically shifted attention for traffic prediction. Du et al. [23] introduced a hybrid method based on multiple 2DCNN-GRU units for short-term traffic flow forecasting, effectively capturing spatial features and long-term temporal dependencies. Chen et al. [11] exploited multiple stacked 3DCNNs to capture low-level spatio-temporal correlation features for vehicle flow prediction. Jiang et al. [24] presented a novel pyramid ConvLSTM architecture model (DeepCrowd) to simultaneously capture spatial and temporal dependency. However, these methods cannot fully explore low-level spatio-temporal correlation and the local region's dependency features.

For heterogeneity of traffic flow, most literature cannot pay attention to this. As early as 2011, Cheng et al. [25] discovered the spatio-temporal autocorrelation feature of traffic data. Correctly recognizing and recalibrating correlations' strength is vital in modelling traffic data. Cen and Chen et al. presented the MST3D and LMST3D-ResNet network [11,13], respectively. They used stacked shared 3D convolution operations to handle spatio-temporal features, implying that the influence in different spatial regions is the same. Guo et al. [12] noticed the heterogeneity and presented an 'RC' block to capture traffic flow's heterogeneity, which led to many parameters.

2.2. 3D Densely connected convolution neural networks

Adequate access to spatio-temporal information in traffic flow is of great significance. The 3D Convolutional Neural Network (3DCNN) model is one of the best models for capturing spatio-temporal features. Tran et al. [26] and Ji et al. [27] found that

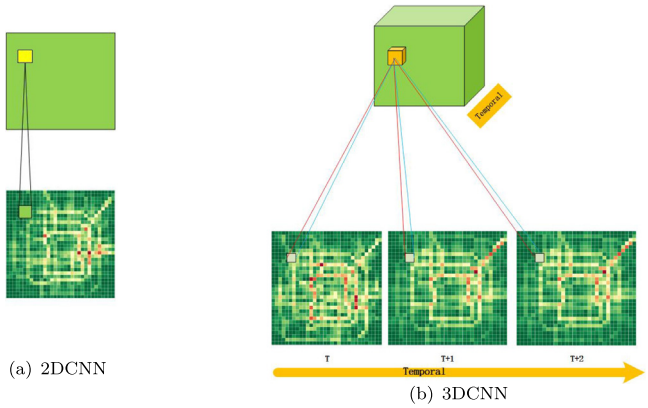


Fig. 2. The comparison of 2D and 3D convolution.

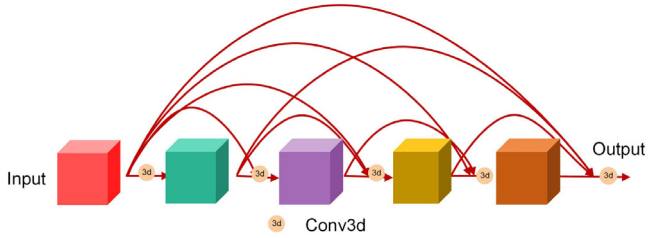


Fig. 3. 3D densenet.

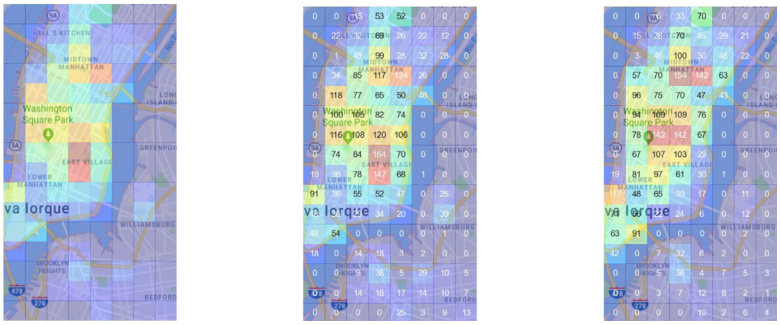
3DCNN is more efficient for capturing both temporal and spatial information than 2DCNN. Fig. 2 displays the comparison results between 2DCNN and 3DCNN. It is observed that the two-dimensional convolutional operations merely can capture local 2D spatial information from the two-dimensional plane in Fig. 2(a). As 2D CNN performs well in learning two-dimension features (latitude and longitude), it is difficult to capture the time information. While a 3D convolution operation is to convolve a 3D filter on a cube generated by stacking multiple contiguous frames, as shown in Fig. 2(b). This can effectively capture motion information because one feature map contains multiple adjacent frames. In other words, 3DCNN can simultaneously deal with both temporal and spatial features owing to preserving the temporal dimension information.

As the observations gained at local spatial locations are correlated, traffic flow indicates local correlation in space. In the traffic flow prediction methods, considering low-level spatio-temporal correlation features and dependency features among local regions, a deeper network needs to be designed to capture spatial dependence fully. Inspired by Zhang and Du's work [28,29], the densenet network in Fig. 3, which performs better than the Resnet [21] network with fewer parameters and computational costs, can strengthen the data flow to capture local spatial correlations among all regions by the more skip connection and feature reuse.

Traffic flow slices at each moment can be regarded as a traffic frame. In this paper, we first exploit a 3D densenet network in traffic forecasting tasks, aiming to model local regional spatio-temporal information encoded in traffic flow automatically.

3. Problem formulation

This section mainly introduces some basic conception of the traffic flow and then elaborates on the core definition and notation about the citywide traffic flow prediction problem.



(a) Grid area about New York City (b) Inflow frame (c) Outflow frame

Fig. 4. Traffic flow processing procedure about New York.

Definition 1 (*Grid Area*$M_{a,b}$)**.** In this study, as in [5], the whole city is regarded as a region based on an actual map by longitude and latitude, covering almost the entire city. The whole map is split into the $A \times B$ region. For instance, as shown in Fig. 4, we divide New York city into 16×8 non-overlapping grid areas.

Definition 2 (*Traffic Frame* $\boldsymbol{M}_{\boldsymbol{a,b}}^{(\boldsymbol{t})}$)**.** We define geographical regions, observe at fixed time intervals (such as 0.5 h or 1 h) and convert traffic data into a two-dimensional grid image $M_{a,b}^{(t)} \in R^{P \times A \times B}$. P, A, and B stand for the image channel, width, and height. In our study, P is fixed as 2, meaning inflow and outflow. $M_{a,b}^{(t)}$ represents each area's traffic flow volume of the traffic frames at some moment t (a and b represent row and column, respectively). Figs. 4(b) and 4(c) show inflow and outflow of the spatial traffic frame image.

Traffic flow GPS information needs to be converted into spatial traffic frames. The specific reasons are outlined as follows:

- The city' traffic flow is predicted by converting GPS information into spatial traffic frame. It is an extensive range of predictions compared to road-based traffic flow prediction.
- Converting the prediction area into a traffic frame can better capture spatial features (for example, local spatial correlation and distant correlation). It is widely known that the classical neural network is an expert in extracting spatial correlation.
- Converting to a traffic frame is crucial for GPS data processing. This scheme may easily convert the GPS information data into a matrix vector, which is a general format for the input of the deep learning model. Some previous works [4,5,24] have demonstrated up-to-date image form outcomes for traffic forecasting.

Definition 3 (*Inflow and Outflow*)**.** Inflow and Outflow refers to the number of vehicles entering (leaving) a specific region in the fixed time interval Δt, the inflow and outflow of the citywide traffic data in each area (a, b) are defined as:

$$M_{in,a,b}^{t} = \sum_{T_r \in U} \left| \left\{ \beta > 1 | \eta_{\beta-1} \notin (a,b) \, \Lambda \eta_{\beta} \in (a,b) \right\} \right| \tag{1}$$

$$M_{out,a,b}^{t} = \sum_{T_r \in U} \left| \left\{ \beta \geq 1 | \eta_{\beta} \notin (a,b) \, \Lambda \eta_{\beta+1} \in (a,b) \right\} \right| \tag{2}$$

where $T_r : \eta_1 \rightarrow \eta_2 \rightarrow \eta_3 \cdots, \rightarrow \eta_{|T_r|}$ is a trajectory in trajectories set U. η_β is the geospatial coordinate, $\eta_\beta \in (a, b)$ denotes the spot η_β in the grid (a, b). $|\cdot|$ means the base of a set.

Definition 4 (*Multiple Temporal Dependencies*)**.** Multiple temporal dependencies include **Closeness, Daily-Period, and Weekly-Period**.

Closeness The traffic data is affected by the recent time interval. E.g., the traffic flow Jam at 10:00 a.m. will affect the traffic flow at 11:00 a.m. Here, if the predicted target is the predicted traffic at 11:00 a.m. on Monday, the length of closeness is four, and the time interval is one hour. Then, the closeness denotes traffic flow at 7:00, 8:00, 9:00, 10:00 a.m.

Daily Period: Traffic conditions in peak hours in the morning and evening are analogous in sequential working days, which repeat every 24 h. Fig. 5 shows an apparent period of traffic flow of four different areas for BikeNYC data in one week.

Weekly Period: The traffic flow at 8:00 a.m. this Monday is similar to last Monday, repeated once a week.

Definition 5 (*Complex Spatial Dependence*)**.** Complex spatial dependencies contain the **Local spatial correlation** and **long-range dependencies**.

Local spatial dependencies: Fig. 6 displays three local spatial regions (such as Living areas $M1$, commercial areas $M2$, university $M3$). For instance, residential and commercial areas may cause substantial changes when people from living areas travel to commercial areas for shopping during the festivals and holidays. The university may cause large changes when students leave school on weekends and go to other places for free activities.

Long-range dependencies: In modernistic cities, the human can go anywhere in a short time by freeway or subway. For instance, humans in living areas $M1$ can go to commercial areas $M2$ for shopping by subway. Similarly, $M4$ and $M5$ are connected by freeways, and their traffic flow can lead to interaction quickly.

Definition 6 (*External Factors* E_{ext}^{t})**.** There are also many external factors, which impact traffic flow forecasting. For example, people prefer to stay at home or indoor public places in hot summer. However, when the temperature is suitable, they prefer shopping or visiting scenic spots on holidays. Our study mainly considers weather information, holidays, and metadata (weekday/weekend).

Meteorological information processing: meteorological information mainly includes weather conditions, temperature, wind velocity, pressure and so on. Classification properties are digitized by One-Hot Encoding [30]. Unclassified properties (temperature and humidity etc.) are scaled into the range $[-1, 1]$.

Holidays and metadata processing: all categories of festivals (such as Christmas, Mother's Day, Easter, etc.) can be obtained from the calendar and encoded into a binary vector.

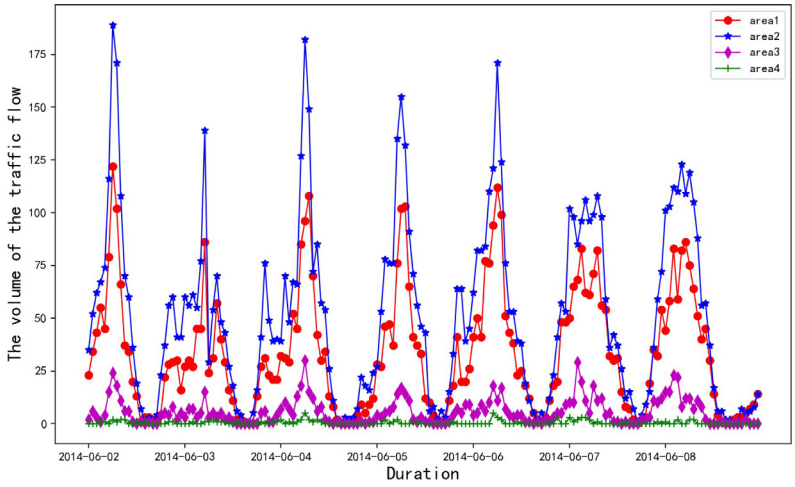


Fig. 5. Four areas of inflow on BikeNYC data for one week.

Fig. 6. Local spatial dependencies and long-range dependencies.

All the external information is combined and fed into the fully connected layers to generate the traffic flow feature, denoted as $E_{ext}^{t} \in R^{2\times A\times B}$.

Question 1 (*Single-Step Citywide Traffic Flow Prediction*)**.** The goal of citywide traffic flow prediction is to forecast inflow and outflow in each given specific area in the next time interval Δt, that is, $\mathrm{R} = \left\{M^{1}, M^{2}, M^{3} \ldots, M^{t-1}\right\}$ represents history observed value, the target is to predict M^{t}.

Question 2 (*Multi-Step Citywide Traffic Flow Prediction*)**.** The core idea is to predict the subsequent interval data by setting predicted data as input data, such as two-step prediction. First, acquire M^{t} by Question 1, then regard M^{t} as the input for the model to forecast the next period M^{t+1}. For the convenience of reading, the notation is defined as shown in Table 1.

4. 3D Densenet multiscale ConvLSTM-Resnet framework

In this section, we propose a new end-to-end framework, called Spatio-Temporal 3D Densenet Multiscale ConvLSTM-Resnet Network (ST-3DDMCRN), to learn traffic frame's spatio-temporal properties. Fig. 7 displays the detail of the whole architecture, which contains three temporal modules (i.e., closeness, daily-period, and weekly-period), external module, and fusion module.

Table 1
Notation description in this paper.

Notation	Description
$M = \{m_{0,0}, m_{0,1}, m_{0,2}, m_{0,3}, \ldots m_{a,b}\}$	The region set of traffic frames
E_{ext}^{t}	The external situation at time t
$M^{t} \in R^{2*a*b}$	Citywide traffic flow at t
t	Available traffic flow states
Δt	Minimum time interval
M^{t}	Predictive traffic frame

The three temporal modules share the same network structure, including the 3D densenet layer, multiscale ConvLSTM-Resnet (M-Resnet) block and Region-Squeeze-and-Excitation (RSE) unit. The current traffic frame is closely related to the recent historical data. As shown in Fig. 5, it is clear that traffic flow has closeness and periodic patterns. Besides, traffic flow also displays some trend patterns with the change of seasons, e.g., once summer arrives, the morning rush hour will be earlier.

The goal of the closeness module is to handle spatio-temporal characteristics of the current historical traffic frames. Let the closeness sequence be as $\mathrm{S} = \left[M^{t-S_c}, M^{t-(S_c-1)}, \ldots, M^{t-1}\right] \in \mathrm{R}^{S_c\times P\times A\times B}$, where S_c represents the length of the sequence. The daily-period and weekly-period modules depict the periodic patterns in traffic flow. Their input comes from the last few days and weeks in traffic flow, which has a similar periodic characteristic as the target to be predicted. The daily-period and weekly-period subsequence are defined as $\mathrm{D} = [M^{t-S_d \cdot d_{daily}}, M^{t-(S_d-1)d_{daily}}, \ldots, M^{t-d_{daily}}]$ and $\mathrm{W} = \left[M^{t-S_w \cdot w_{weekly}}, M^{t-(S_w-1)w_{weekly}}, \ldots, M^{t-w_{weekly}}\right]$, where the d_{daily} is set to one day, S_d represents the length of the daily-period sequence, the w_{weekly} is set one week, S_w means the length of the weekly-period sequence. Similarly, they also utilizes 3D densenet and M-Resnet blocks, and an RSE unit to capture period patterns. At the end of each temporal module, a ConvLSTM lay is applied to return the final output. Eventually, the fusion module combines the predictions based on the temporal modules and the external factors to obtain a final prediction as M_s^t.

(1) Closeness module:

(a) 3D densenet block: In the traffic domain, traffic frames at contiguous time intervals are associated among all regions (low-level spatio-temporal dependencies). To make full use of 3D convolutional neural networks to capture feature information from local spatial regions, we increase the depth of the network as much as possible. Nevertheless, stacked 3DCNN could easily lead to gradient diffusions or gradient explosions. To overcome

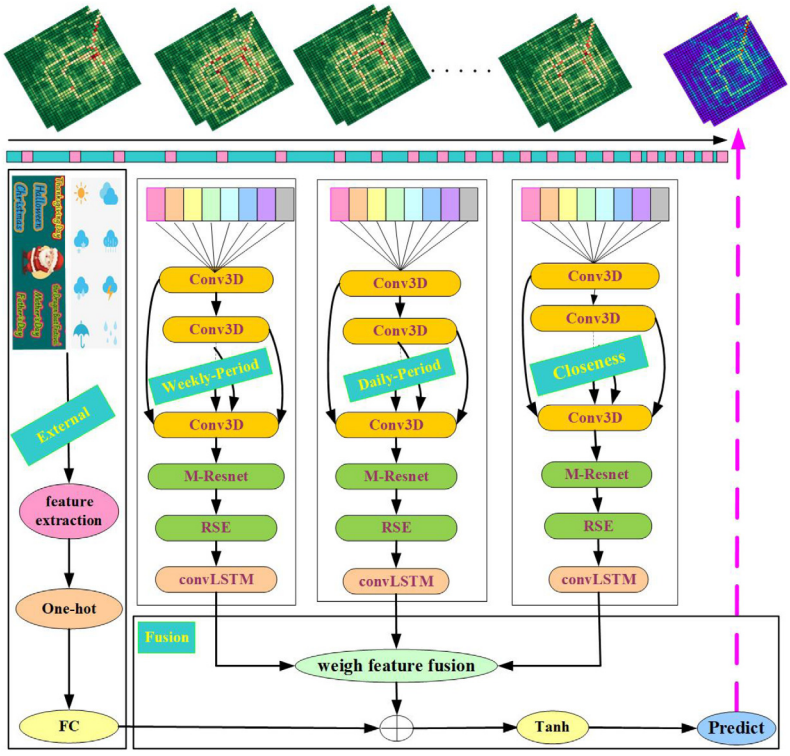


Fig. 7. ST-3DDMCRN architecture. M-Resnet: Multiscale ConvLSTM-Resnet, RSE: Region-Squeeze-and-Excitation Networks.

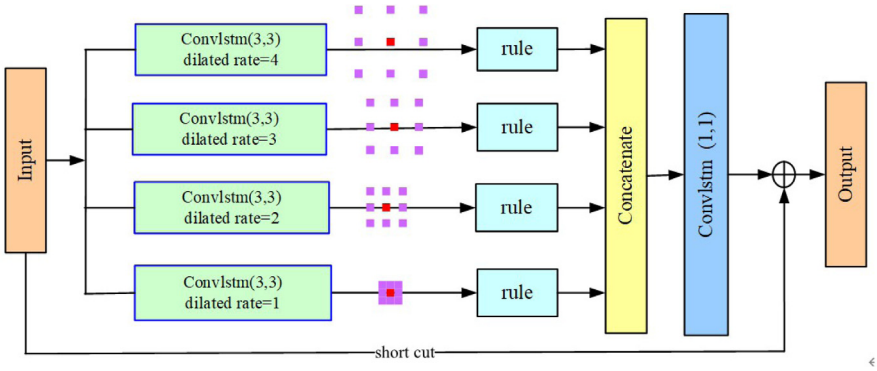


Fig. 8. Dilated residual ConvLSTM unit.

the problem, we propose the 3D densenet network to obtain a more deeper network structure. Formally,

$$M_s^{L_s} = F_u(M_s^0 \oplus M_s^1 \oplus \cdots \oplus M_s^{L_s-1}), L_s = 1, 2, \ldots, l \tag{3}$$

where $M_s^{L_s} \in R^{S_c \times C_s \times A \times B}$, C_s denotes the number of channels. F_u denotes a non-linear transform function that includes three operations (3DCNN, Batch Normalization (BN) and Hyperbolic Tangent Function (Tanh)). $\oplus$ means the concatenation of the future maps produced in all 3D convolution lays. By applying the 3D densenet network to multiple contiguous traffic frames, feature maps have obtained the traffic flow's low-level spatio-temporal correlation features and local spatial correlation. Then, the initial input is denoted as M_s^0, with the dimension $S_c \times C_s \times A \times B$, the densely connected pattern is used in the 3DCNN network to improve feature propagation and prediction efficiency, the result after applying the 3D densenet block would be $S_c \times C_s \times A \times B$.

(b) M-Resnet block: Long-range spatial correlation among all regions in the transportation systems play an essential role. Previous works [4,5,24,31] failed to capture long-range spatial dependencies by stacked multi-layers convolutional networks. Therefore, we design an M-Resnet block as shown in Fig. 8, containing four parallel dilated ConvLSTM layers with different dilated rates. Then, these four paths are concatenated and then fused by one 1×1 ConvLSTM layer, which includes different scales characteristics, to generate the final feature map. It is worth noting that four paths are individually processed with the same fixed convolutional kernel and diverse dilated rate, which can effectively capture various receptive fields without losing spatial information. The residual mapping is defined as Φ. Formally,

$$M_s^{L_s+l} = M_s^{L_s+l-1} + \Phi(M_s^{L_s+l-1}; \gamma_c^{(l)}), \qquad l = 1, 2, 3, \ldots \tag{4}$$

where $M_s^{L_s+l-1}$ is the input of the lth residual unit, $M_s^{L_s+l} \in R^{S_c \times C_s \times A \times B}$, C_s denotes the number of channels, $\gamma_c^{(l)}$ is the learnable parameters in the Lth residual unit. Then, the initial input of the M-Resnet block is denoted as $M_s^{L_s}$, with the dimension $S_c \times C_s \times A \times B$. Then, the network can learn the different receptive features by Concatenate operation, and reduce the feature map with the same dimension as the input by one 1×1 ConvLSTM layer. The output of the M-Resnet block is $S_c \times C_s \times A \times B$.

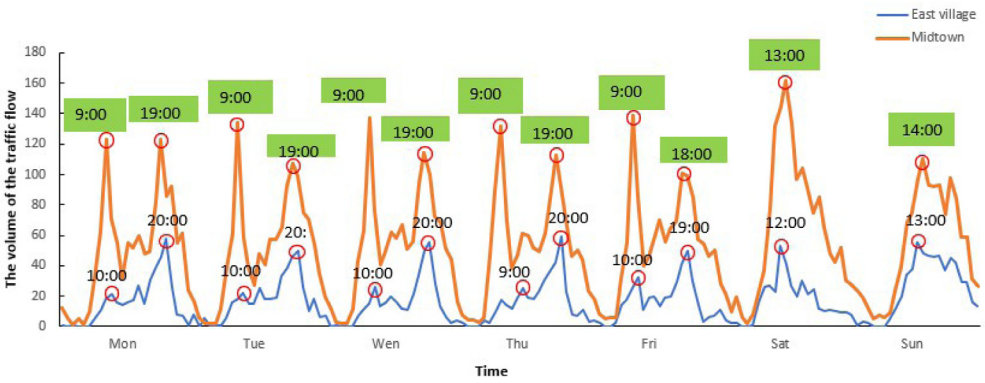


Fig. 9. Inflow on bicycle trajectory data at East Village and Midtown for a week.

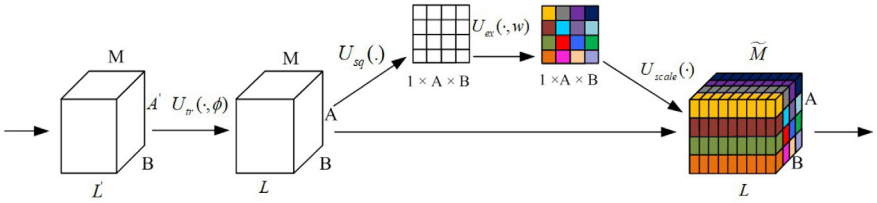


Fig. 10. The structure of the RSE unit.

(c) RSE unit: It is worth noting that the contribution of adjacent regions to the prediction is different across the whole space. If we do not consider heterogeneity, it will not be adequate to simulate the correlation of traffic flow.

Take the bicycle flow in New York City as an example. We pick up East Village and Midtown in New York, and observe the traffic flow volume, as shown in Fig. 9. On the one hand, it is noticed that the traffic flow shows apparent cyclical patterns due to the impact of daily human routine. On the other hand, the rush hours are not exactly at the same time for the East Village and Midtown. In other words, the strength of temporal periodicity also varies with different regions.

Then, recognizing and recalibrating the varying extent of the feature's role in space is vital for precisely simulating spatio-temporal features. Inspired by the Squeeze-and-Excitation (SE) network [32] in feature extraction, we propose the RSE unit, which quantifies the extent of contributions of channel-wise features for each region to improve our model's capacity. Fig. 10 displays the architecture of the RSE unit, which includes squeezing and excitation operation. RSE unit firstly calculates the average value of the same region in different channels (features) by average pooling operation, and then compresses and obtain a spatial distribution map Q as follows:

$$Q = U_{sq}(\sum_{k} m_k^i) = \frac{1}{L}\sum_{k=1}^{L} m_k^i \qquad i = 1, 2, \ldots, A \times B \tag{5}$$

where $Q \in R^{A\times B}$, $L = S_c \times C_s$, i represents the region index in a single-channel feature map, k refers to the channel index. U_{sq} is the operation to compute each region's average in different channels. Then, we perform an excitation operation and select the same Sigmoid function as the SE network:

$$O = U_{ex}(Q, X) = \sigma(g(Q, X)) = \sigma(X_2 \odot \delta(X_1 \odot Q)) \tag{6}$$

where X_1 and X_2 are the learned parameters. $\odot$ is the operation of convolution. σ and δ denote Sigmoid and Tanh function, respectively. Then, we make a spatial elementwise multiplication between the input $M_s^{L+l_s}$ and the weight feature map O in different region index, and the multiplication result is denoted as $\widetilde{M}_s$. The calculation function is as follows:

$$\widetilde{M}_s = U_{scale}(M, O) = M_s^{L_s+l} \otimes O \tag{7}$$

where the output $\widetilde{M}_s = [\widetilde{m}_1, \widetilde{m}_2, \widetilde{m}_3, \ldots, \widetilde{m}_L]$, $\widetilde{m}_j = [\widetilde{m}_j{}^1, \widetilde{m}_j{}^2, \widetilde{m}_j{}^3, \ldots, \widetilde{m}_j{}^{A\times B}]$, $O = [o^1, o^2, o^3 \ldots o^{A\times B}]$. U_{scale} is the spatial elementwise multiplication operation, which multiplies the element m^i at the same position in different channels with the element o^i, located in the corresponding position in O.

The implementation details of the RSE unit are shown in Fig. 11. The $M_s^{L_s+l} \in R^{S_c\times C_s\times A\times B}$, the first step is a position-squeeze procedure using an average pooling operation. It leverages the spatial features $M_s^{L_s+l}$, which has a dimension of $L \times A \times B$, ($L = S_c \times C_s$) by the reshape operation, to calculate the average for each position $U_{sq}(.)$ in space, which is $1 \times A \times B$. The second step is the Excitation operation, which assists the network in learning about the dependency $U_{ex}(., w)$ between the features and the location, then adjusting the feature map $U_{scale}(.)$ based on the dependency. The adjusted feature map M_s is the RSE unit's output, with a dimension of $L \times A \times B$.

Therefore, the difference of spatial feature contribution is sufficiently considered in the RSE unit. That is to say, the RSE unit achieves each area's feature recalibration. At the end of the closeness module, a ConvLSTM lay is utilized to obtain the final output:

$$M_s^t = ConvLSTM(\widetilde{M}_s) \tag{8}$$

where $M_s^t \in R^{P\times A\times B}$, P takes the value of 2, indicating the inflow and outflow.

(2) Daily period and weekly period modules: Traffic flow usually displays obvious periods and trends due to the regular daily routine of people. According to this, the daily and weekly modules have the same architecture as the closeness module. 3D densenet is used in the daily and weekly modules to learn the traffic flow's periodic and trend characteristics. Formally,

$$M_d^{L_d} = F_u(M_d^0 \oplus M_d^1 \oplus \cdots \oplus M_d^{L_d-1}), L_d = 1, 2, \ldots, l \tag{9}$$

$$M_w^{L_w} = F_u(M_w^0 \oplus M_w^1 \oplus \cdots \oplus M_w^{L_w-1}), L_w = 1, 2, \ldots, l \tag{10}$$

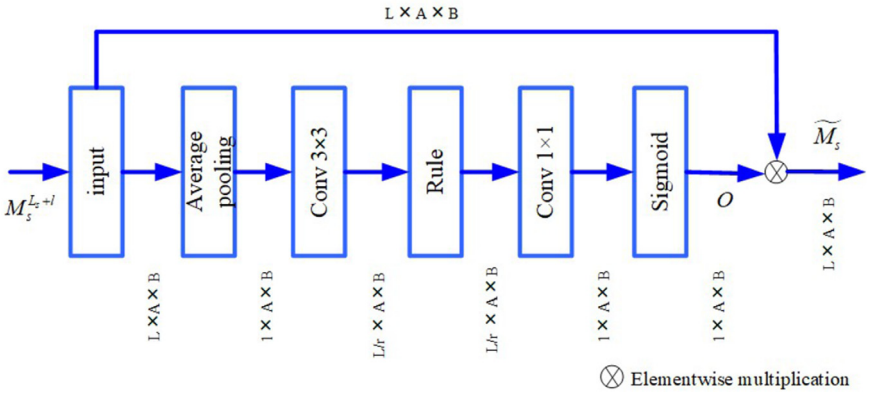


Fig. 11. Basic structure of RSE unit.

where $M_d^l \in R^{C_d \times S_d \times A \times B}$, $M_w^l \in R^{C_w \times S_w \times A \times B}$, C_s and C_w denote the number of channels in 3D convolution in daily-period and weekly-period, respectively. F_u denotes a function that includes three consecutive operations, including 3D CNN, BN, and Tanh. $\oplus$ means the concatenation operation. Afterwards, M-Resnet block are utilized to capture complex spatial feature in daily period and weekly period. The residual mapping is denoted as Φ. Formally,

$$M_d^{L_d+l} = M_d^{L_d+l-1} + \Phi(M_d^{L_d+l-1}; \gamma_d^{(l)}), \qquad l = 1, 2, 3, \ldots \tag{11}$$

$$M_w^{L_w+l} = M_w^{L_w+l-1} + \Phi(M_w^{L_w+l-1}; \gamma_w^{(l)}), \qquad l = 1, 2, 3, \ldots \tag{12}$$

where $M_d^{L_d+l} \in R^{S_d \times C \times A \times B}$, $M_w^{L_w+l} \in R^{S_w \times C \times A \times B}$, $\gamma_d^{(l)}$ and $\gamma_w^{(l)}$ are the set of the learnable parameters in residual unit. $M_d^{L_d+l-1}$ and $M_w^{L_w+l-1}$ represent the input of the L residual unit.

Subsequently, an RSE unit is applied to recalibrate feature weights for different regions and we obtain the last output by ConvLSTM lay:

$$M_d^t = ConvLSTM(U_{scale}(M, O)) = ConvLSTM(M_d^{L_d+l} \otimes O) \tag{13}$$

$$M_w^t = ConvLSTM(U_{scale}(M, O)) = ConvLSTM(M_w^{L_w+l} \otimes O) \tag{14}$$

where $M_d^t \in R^{2 \times A \times B}$, $M_w^t \in R^{2 \times A \times B}$.

(3) Fusion module: We display how to fuse the three temporal and external module in this part. Fig. 7 shows that the output of the three temporal modules is indicated as M_s^t, M_d^t and M_w^t, respectively. There is a global difference between the extent to the closeness, the daily-period, and weekly-period modules. Long-term period patterns are apparent for some regions, but the current closeness feature's influence is more crucial for other areas. Therefore, we fuse the three outputs mentioned above through a linear weighting method, then obtain an integrated spatio-temporal tensor as follows:

$$\widetilde{M}_F = W_{closeness} \circ M_s^t + W_{daily-period} \circ M_d^t + W_{weekly-period} \circ M_w^t \tag{15}$$

where $W_{closeness}$, $W_{daily-period}$, and $W_{weekly-period}$ mean learnable parameters denoting three modules' contributions for the forecasting target. $\circ$ represents the multiplication of the corresponding coordinate of values. $\widetilde{M}_F \in R^{2 \times A \times B}$ denotes the prediction results.

As traffic flow is affected by some external factors, including weather situations, holidays, and metadata (such as weekday/weekend). To improve the prediction performance further, we encode the external information as $\widetilde{E_{ext}}$ by Definition 6. Finally, a tanh activation layer is applied to fuse the two outputs as follows:

$$\widehat{M^n} = \tanh(W_F \circ \widetilde{M}_F + W_{ext} \circ \widetilde{E_{ext}}) \tag{16}$$

where W_F and W_{ext} are the parameter matrices. $\circ$ is the element-wise multiplication. The hyperbolic tangent tanh ensures the output values are within the range $[-1, 1]$. Furthermore, the predicted $\widehat{M^n} \in R^{2 \times A \times B}$ is rescaled back to the normal value by the inverted min–max linear normalization. Finally, our model is trained through minimizing the Mean Squared Error (MSE) between the actual values and the predicted values:

$$\ell_\varphi = \left\| M^t - \widehat{M^t} \right\|_2^2 \tag{17}$$

where φ represents the learnable parameters in ST-3DDMCRN.

(4) Model Training: Algorithm 1 demonstrates the algorithmic procedure of our ST-3DDMCRN model. The historical traffic flow is divided by different temporal features (closeness, daily-period, and weekly-period) (lines 1–7). Then, input all data, train the ST-3DDMCRN model and choose Adam optimization [33] as the optimizer (lines 8–13). The learned ST-3DDMCRN model will be sent back (line 14).

Algorithm 1 Training procedure of ST-3DDMCRN Algorithm.

Require:
Historical observations of traffic frame: $R = \{M^0, M^1, M^2 \ldots M^{t-1}\}$
External information: $E_{ext} = \{E_{ext}^0, E_{ext}^1, E_{ext}^2 \ldots E_{ext}^{t-1}\}$
Lengths of closeness, daily-period, weekly-period: S_c, S_d, S_w
Target variables: M^t
Ensure: Learned ST-3DDMCRN Model

1: $\Psi_{train} \leftarrow \emptyset$
2: **for** $t \in T$ **do** // T denotes all available time-series segments.
3: $S = \left[M^{t-S_c}, M^{t-(S_c-1)}, \ldots, M^{t-1}\right]$
4: $D = \left[M^{t-S_d \cdot d_{daily}}, M^{t-(S_d-1)d_{daily}}, \ldots, M^{t-d_{daily}}\right]$
5: $W = \left[M^{t-S_w \cdot w_{weekly}}, M^{t-(S_w-1)w_{weekly}}, \ldots, M^{t-w_{weekly}}\right]$
6: Put each instance S, D, W, E_{ext} into the corresponding module. $//M^n$ is the target
7: **end for**
8: //training this model
9: Initialize the parameters φ in ST-3DDMCRN
10: **repeat**
11: Randomly select a batch of instances Ψ_{batch} from Ψ_{train}
12: Find φ by minimizing the loss function (17) with Ψ_{batch}
13: **until** maximum epoch or early stopping criteria is satisfied
14: **Return** the learned ST-3DDMCRN model

In Algorithm 1, R, E_{ext} are the input of our ST-3DDMCRN model, Ψ_{train} means the training set. Training instance is constructed from traffic data. Afterwards, backpropagation is continually used to train the model.

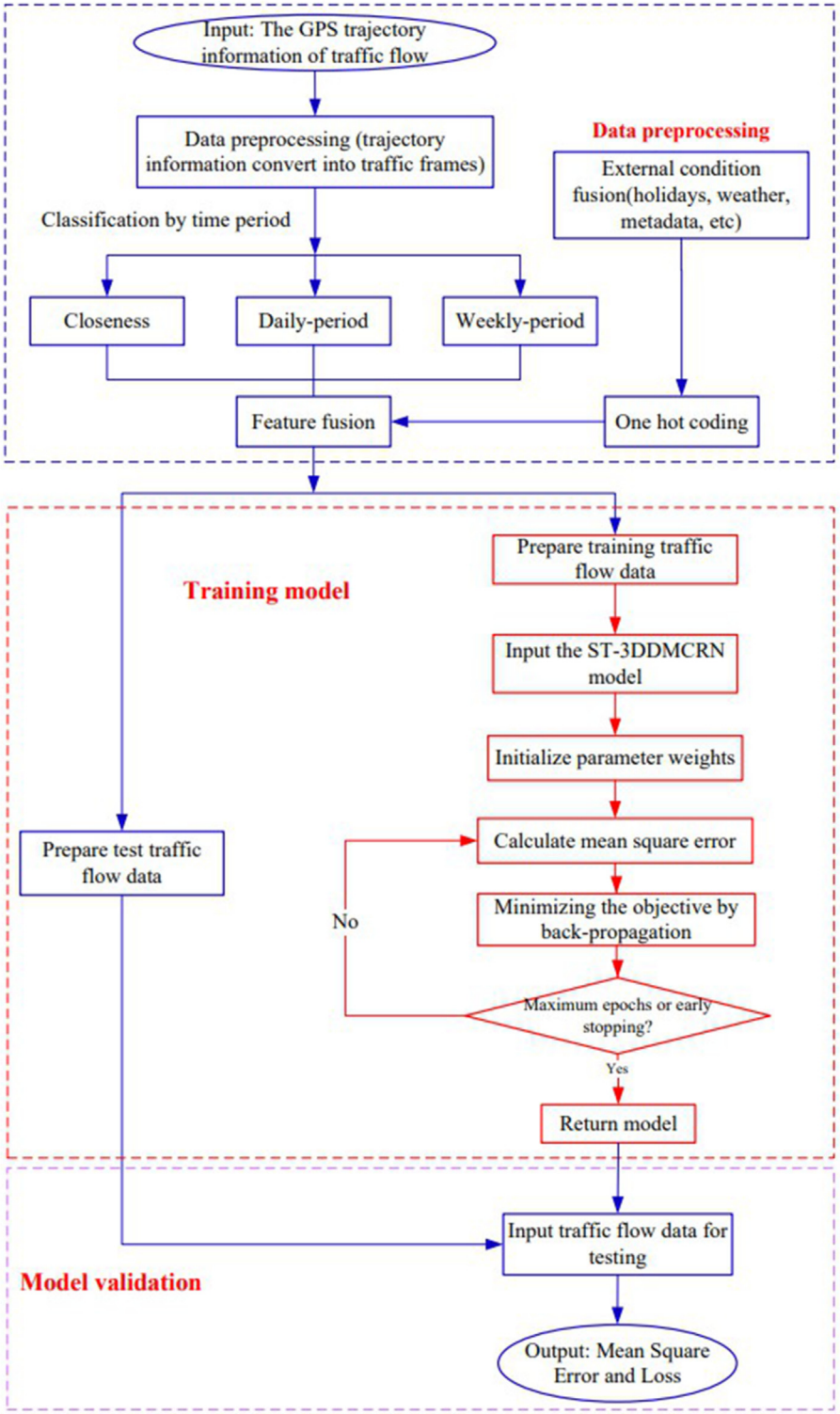


Fig. 12. Flow chart of traffic flow prediction.

To better understand the whole prediction process, the flow chart of the ST-3DDMCRN is shown in Fig. 12. Firstly, trajectory data is collected from GPS devices, then converted into traffic frames. Secondly, put three periods of traffic data into the training model with external information. Then, the traffic data is split into the training and test samples, and establish the ST-3DDMCRN model (see Fig. 7). Lastly, we exploit the trained parameters to verify the experimental results of the test set.

5. Experimental result and analysis

In this section, we first introduce the commonly-used dataset and evaluation metrics of citywide traffic flow prediction. Then, we compare the proposed method with several baseline models. Moreover, to prove the universality of our proposed method, we exploit the model to predict the passenger pickup/dropoff demand tasks.

Table 2
The overview of BikeNYC and TaxiBJ datasets.

Dataset		BikeNYC	TaxiBJ
Traffic flow	Urban	NewYork	Beijing
	Data Format	Bike Rent	Taxi GPS
	Time	S1:2014.04.01–2014.09.30	S1:2013.07.01–2013.10.30 S2:2014.03.01–2014.06.30 S3:2015.03.01–2015.06.30 S4:2016.11.01–2016.04.10
	Time interval	60 min	30 min
	Grid Map Size	(16, 8)	(32, 32)
	Time Intervals	4392	22459
	Bike/Taxi quantity	6800+	34000+
External items	Vacation	20	41
	Weather Conditions	/	16 types
	Temperature/°C	/	[−24.6, 41.1]
	Wind Velocity/mph	/	[0, 48.6]

Table 3
The meteorological information of BikeNYC datasets.

Date	1st.Apr.2014–30th.Sep.2014
Temperature/°C	0 °C–33.88 °C
wind direction	N, NE, NNE, ENE.etc
Humidity	14%–100%
pressure/in	29.38 in–30.72 in
Dew Point/F	11 F–75 F
wind speed/ mph	0 mph–33 mph
wind gust/mph	0 mph–48 mph
Weather condition	Sunny/Cloudy/Snowy/Windy/Rainy.etc

5.1. Datasets description and configuration

To evaluate the proposed model, we utilize two real-world representative trajectory datasets (BikeNYC and TaxiBJ [5]), which contain traffic flow information throughout the whole city. The detailed description is shown in Table 2.[2]

TaxiBJ Dataset. The dataset includes more than 34,000 taxis and 22,459 time intervals, generated from the Beijing taxis GPS trajectory data. The time interval is half an hour, and the grid size is 32 × 32. The auxiliary information contains temperature, wind velocity, 41 kinds of the festival, and 16 types of weather conditions. The dataset is officially divided into training and test sets. Specifically, the last 28 days are used as the test set, the remaining data is taken as the training set.

BikeNYC Dataset.[3] The dataset contains New York bicycle trajectories with 4392 time intervals from 1st Apr. to 30th Sept. 2014. The time interval is 60 min, and the grid size is 16 × 8. The auxiliary information merely contains holiday information. This dataset has a similar training-test ratio of around 6% as the TaxBJ dataset. The last ten days are used as the test set, and all remaining data is taken as the training set.

Note that the official BikeNYC dataset does not offer meteorological information. We collected the meteorological information from the popular weather web[4] and published it on my cloud disk.[5] Its summary is shown in Table 3.

Table 4
Server configuration.

Item	Parameter
Operating system	Linux Ubuntu16.04 LTS
Memory	128 G
CPU	Intel i9 9900 k 3.6 GHz
GPU	GeForce RTX 2080ti
Language	python 3.6

Configuration The Keras[6] (2.2.4) and TensorFlow[7] (1.13.1) are adopted to implement the ST-3DDMCRN model. The other environment configuration details are shown in Table 4,

5.2. Implementation details

5.2.1. Data preprocessing

In the preprocessing part, two kinds of traffic flow are obtained from the TaxiBJ and BikeNYC datasets by Definitions 2 and 3. For the external features, we convert them into a binary vector by Definition 6. Min–max normalization is performed on all these methods to improve the model's performance and convergence speed. The definition of the min–max transformation is as follows:

$$m^*_{i,j} = \frac{m_{i,j} - m_{\min}}{m_{\max} - m_{\min}} \times 2 - 1 \tag{18}$$

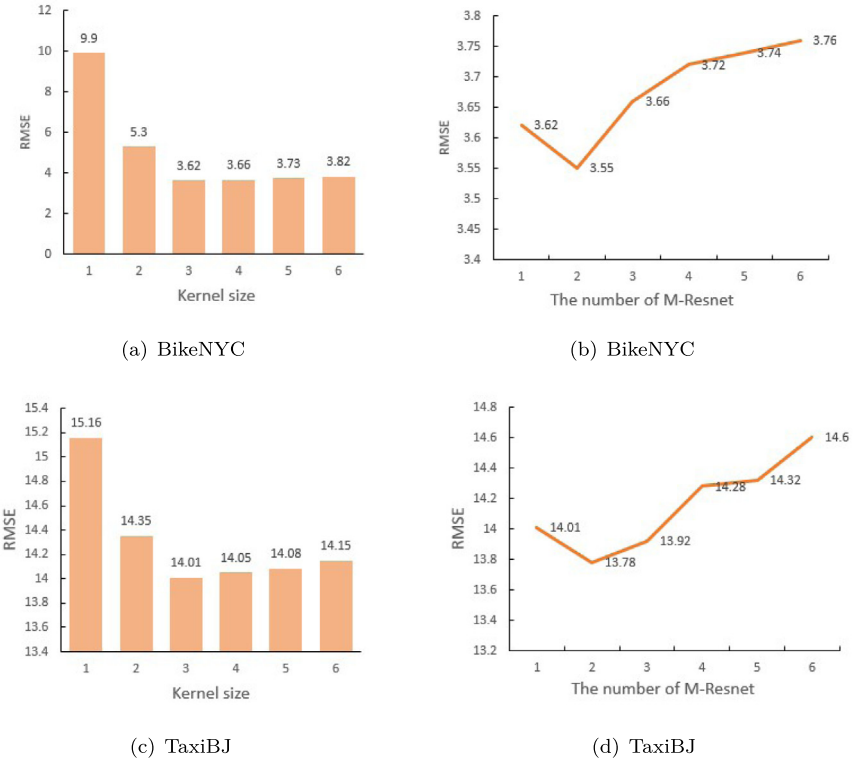
where $m_{i,j}$ represents the original value of the initial data, $m_{\max}$ and $m_{\min}$ denote the maximum and the minimum values of the sample data, respectively, and $m^*_{i,j}$ is the transformed result between $[-1, 1]$. When we obtain the model's predicted value $m^*_{i,j}$, we rescale them to generate the actual value $m_{i,j}$. For fair comparison, we evaluate various comparison algorithms on the same testing sets.

5.2.2. Evaluation metric

The performances of all methods are evaluated with root mean square error (RMSE) and mean absolute error (MAE), which are common-used evaluation metrics in traffic prediction tasks. Specifically, they are defined as:

$$RMSE = \sqrt{\frac{1}{\lambda}\sum_{y=1}^{\lambda}\left(\widehat{Z}_y - Z_y\right)^2} \tag{19}$$

[2] The details of BikeNYC and TaxiBJ dataset are quoted from [5].
[3] https://www.citibikenyc.com/system-data
[4] https://www.wunderground.com
[5] https://www.jianguoyun.com/p/DaMGJ7YQzOu0CRisnf4D
[6] http://keras.io
[7] http://www.tensorflow.org



(a) BikeNYC (b) BikeNYC

(c) TaxiBJ (d) TaxiBJ

Fig. 13. Effect of different hyperparameters configurations on the BikeNYC and TaxiBJ datasets.

$$MAE = \frac{1}{\lambda}\sum_{y=1}^{\lambda}\left|\widehat{Z_y} - Z_y\right| \tag{20}$$

where λ is the total number of samples used for validation, $\widehat{Z_y}$ and Z_y represent the predicted and real value, respectively.

5.2.3. Tuning hyperparameters

In this experiment, the 3D densenet block includes three submodules, and each submodule includes a 3D CNN, BN and Tanh operation, 3D CNN use $l \times d \times d$ kernel size, where l equal to the size of the input data in the temporal dimension. The S_c, S_d, and S_w of closeness, daily-period, weekly-period are 3, 2, 2 for the BikeNYC dataset, respectively. The S_c, S_d, and S_w of closeness, daily-period, weekly-period are 4, 3, 3 for the TaxiBJ dataset, respectively. The channels are 16 and 32 for the BikeNYC and the TaxiBJ dataset. The fully-connected layers and all convolutional layers are initialized by Xavier [34]. Tanh is used as all the activation functions in the model. The initial learning rate is set as 0.0005 and updated constantly by the ReduceLROnPlateau callbacks function. The maximum training epoch and minibatch size are 400 and 64, respectively. We optimize the network parameters via the Adam optimization [33] and utilize an early-stopping mechanism to prevent the model from overfitting.

To explore the effect of different network configurations, we evaluate the model on the BikeNYC and TaxiBJ datasets by varying the two most critical hyperparameters in Fig. 13. i.e., the kernel size and the number of M-Resnet. Firstly, set the number of M-Resnet blocks to 1 and change the size of the convolution kernel from 1×1 to 6×6. As shown in Figs. 13(a) and 13(c), when the convolution kernel size increases from 1×1 to 3×3, the RMSE declines sharply, and while the kernel size increases from 3×3 to 6×6, the RMSE does not change much in the BikeNYC and TaxiBJ datasets. The larger the convolution kernel leads to a longer training time, the kernel size is set to 3×3. Figs. 13(b) and 13(d) exhibit the influence of M-Resnet unit's numbers on BikeNYC and TaxiBJ datasets. Similarly, the filter size is set as 3×3. It is also clearly shown that when the number of M-Resnet units is equal to 1, the model performance exceeds all the baseline approaches. When the number of M-Resnet units increases to 2, the RMSE reaches the minimum value for the BikeNYC and TaxiBJ datasets, demonstrating the RMSE can be minimized with only two residual units.

5.3. Methods comparison

We divide the spatio-temporal prediction methods into the conventional time-series methods and deep-learning methods, and compare ST-3DDMCRN with the following baselines:

5.3.1. Baseline models

Conventional Time-series Models:

- **HA**: Historical Average (HA) is a basic sequence prediction model, which forecasts future traffic flow by calculating historical average values.
- **ARIMA** [17]: ARIMA is a classical time series model that predicts the future values by manual parameter adjustment.

Deep Learning Model:

- **ST-Resnet** [5]: The ST-Resnet[8] model is a deep neural network that employs multiply stacked standard Resnet units to capture spatio-temporal correlation by closeness, period and trend.
- **DeepSTN+** [22]: DeepSTN+[9] model can better capture spatio-temporal correlation through the ConvPlus unit and early fusion mechanism.

8 https://github.com/amirkhango/DeepST
9 https://github.com/FIBLAB/DeepSTN

Table 5
Comparison between our model and baselines.

Model	Temporal[a]	Spatial[b]	Spatio-temporal[c]	Local spatial[d]	Long-range[e]	Heterogeneity[f]
HA	✓					
ARIMA	✓					
ST-Resnet	✓	✓				
DeepSTN+	✓	✓				
STDN	✓	✓				
ST-3Dnet	✓	✓	✓			✓
LMST3D-Resnet	✓	✓	✓	✓		
ST-3DDMCRN	✓	✓	✓	✓	✓	✓

[a]Denote the temporal correlation of historical traffic flow.
[b]Denote the spatial correlation of historical traffic flow.
[c]Denote the correlations of traffic flow in both spatial and temporal dimensions simultaneously.
[d]Denote local spatial correlation between nearby locations.
[e]Denote long-range spatial dependence among regions.
[f]Denote the contributions of the correlations both in space and time.

Table 6
Comparison of ST-3DDMCRN with other baselines on BikeNYC AND TaxiBJ.

Method	BikeNYC		TaxiBJ	
	RMSE	MAE	RMSE	MAE
HA	20.32	10.21	44.32	22.60
ARIMA	10.23	5.35	22.73	16.21
ST-Resnet	6.37	2.97	16.69	9.52
DeepSTN+	6.21	2.39	18.04	10.11
STDN	5.78	2.36	16.42	9.41
ST-3Dnet	5.80	2.43	16.09	9.33
LMST3D-Resnet	5.64	2.33	15.67	9.21
ST-3DDMCRN[a]	3.61	1.53	14.15	8.80
ST-3DDMCRN	3.55	1.49	13.85	8.26

[a]Denote the model does not consider external factors.

Table 7
Inflow and outflow results on BikeNYC.

Method	Inflow		Outflow	
	RMSE	MAE	RMSE	MAE
HA	20.02	10.05	20.71	10.41
ARIMA	9.97	5.21	10.47	5.47
ST-Resnet	6.08	2.83	6.63	3.08
DeepSTN+	5.97	2.29	6.57	2.52
STDN	5.51	2.24	6.25	2.55
ST-3Dnet	5.43	2.27	6.21	2.60
LMST3D-Resnet	5.63	2.32	5.66	2.34
ST-3DDMCRN[a]	3.55	1.50	3.64	1.54
ST-3DDMCRN	3.48	1.45	3.67	1.53

[a]Denote the model does not consider external factors.

- **STDN** [6]: The STDN[10] model learns dynamic spatial correlation and consecutive temporal correlation with external factors by uniting CNN and LSTM layers for traffic prediction.
- **ST-3Dnet** [12]: ST-3Dnet[11] model, which first introduces 3D convolution into the traffic field, demonstrates advanced performance on the BikeNYC and TaxiBJ.
- **LMST3D-Resnet** [13]: LMST3D-Resnet[12] utilizes a 3D-ResNet model to better deal with low-level spatio-temporal information for region-based prediction in intelligent cities.

We briefly compare our model with other baselines regarding temporal, spatial, spatio-temporal, local spatial dependencies, long-range spatial dependencies and heterogeneity (see Table 5).

10 https://github.com/tangxianfeng/STDN
11 https://github.com/guoshnBJTU/ST-3DNet
12 https://github.com/deerta0103/LMST3DResnet

5.3.2. Performance comparison with baselines

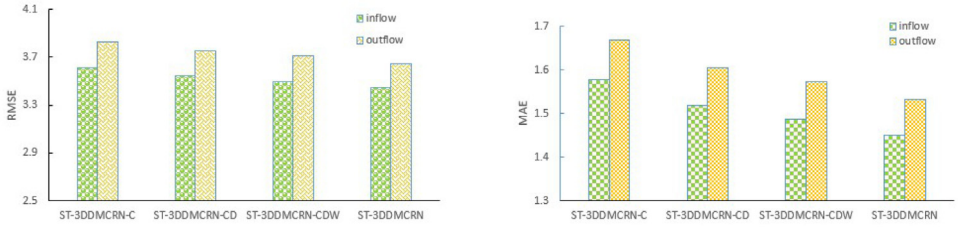
This section compares the ST-3DDMCRN model with other baseline methods on TaxiBJ and BikeNYC datasets in Table 6. Table 7 displays the detailed results of inflow and outflow for BikeNYC in RMSE and MAE. We run each method 10 times and report the mean output of each model.

Among these models, HA and ARIMA, traditional time series methods, merely capture temporal characteristics, performing poorly on these two datasets. The recent DeepSTN+ and ST-Resnet models simulate citywide spatio-temporal correlation and decrease error to some extent by closeness, period, and trend modules. Due to the deep representation learning, they obtain a better performance than traditional time series methods. For example, DeepSTN+ reduces the RMSE to 6.21 on BikeNYC and 18.04 on TaxiBJ. However, both of them ignore the low-level spatio-temporal correlation. Owing to the combination of 2D CNN and LSTM units, the STDN model can explicitly simulate long-term temporal and dynamic spatial dependencies by a flow gating mechanism and periodically shifted attention mechanism. Thus, STDN decreases the RMSE to 16.42 on TaxiBJ and 5.78 on BikeNYC. Nevertheless, it notices temporal features in the high-level layers, and the low-level spatio-temporal correlation cannot be fully exploited. ST-3Dnet model observes the low-level spatio-temporal correlation and heterogeneity of traffic data. LMST3D-Resnet utilizes 3DCNN residual networks to capture the low-level spatio-temporal feature and local regional spatial correlation in intelligent cities. However, they do not consider low-level spatial–temporal correlation, multiscale spatial correlation, and heterogeneity simultaneously.

In contrast, it is discovered from Tables 5 and 6 that our ST-3DDMCRN model achieves the lowest RMSE and MAE by capturing spatio-temporal correlation, complex spatial correlation and heterogeneity of traffic flow. Specifically, our method on the BikeNYC dataset reduces RMSE to 3.61 without considering external factors, outperforming the previous best approach LMST3D-Resnet by 12.4% relatively. Considering external factors, ST-3DDMCRN achieves better results than ST-3DDMCRN*. Meanwhile, our model shows the highest prediction performance on the TaxiBJ dataset, reducing the RMSE from 15.67 to 13.88, which verifies our model's better performance.

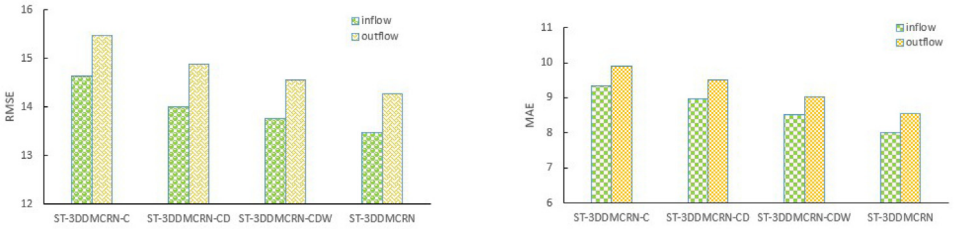
5.3.3. Ablation study

An ablation study on ST-3DDMCRN is described in this section, with variations in the temporal modules and network architecture analysed. First, we study the effectiveness of multiple temporal modules and external modules. Then, we verifies the impact and significance of each core block in multiple temporal modules. The different temporal module's variants are depicted as follows:



(a) RMSE result (b) MAE result

Fig. 14. Different modules results for ST-3DDMCRN on BikeNYC.



(a) RMSE result (b) MAE result

Fig. 15. Different modules results for ST-3DDMCRN on TaxiBJ.

- **ST-3DDMCRN-C**: In this variant, we merely utilize the closeness module to capture the spatio-temporal feature of the traffic flow.
- **ST-3DDMCRN-CD**: This framework exploits the closeness and daily-period modules to capture the spatio-temporal feature of the traffic flow.
- **ST-3DDMCRN-CDW**: This framework considers the closeness, daily-period, and weekly-period of the traffic flow.
- **ST-3DDMCRN**:Our proposed model contains closeness, daily-period, weekly-period and external modules.

Figs. 14 and 15 illustrate the RMSE and MAE of ST-3DDMCRN and their variants on the BikeNYC and TaxiBJ datasets. It can be seen that ST-3DDMCRN-C can apply the closeness module to surpass other baseline methods for traffic flow forecasting. By adding daily-period and weekly-period modules, RMSE and MAE are further decreased. ST-3DDMCRN-CDW demonstrates that exploring periodic correlation can better obtain the spatio-temporal correlation. Adding the external module can further improve the performance, which verifies the external module's effectiveness. Our model merges the closeness, daily-period, weekly-period, and external factors, which can help enhance prediction accuracy.

Due to space constraints, the study focuses on the BikeNYC case study and does not involves the full combinatorics of all possible variants of the proposed model but aims to assess the impact on important modules choices (3D densenet unit, M-Resnet unit, RSE unit), while maintaining all other conditions. More specifically, ST-3DDMCRN is compared to the four different variations described below:

M-Resnet+RSE:We merely employ the M-Resnet block and RSE unit to predict traffic flow in three temporal modules.

3D densenet+RSE: This variant directly concatenates a 3D densenet block and an RSE unit in three temporal modules to predict traffic flow in three temporal modules.

3D densenet+M-Resnet:We directly concatenate 3D densenet and M-Resnet to predict traffic flow in three temporal modules.

Table 8
The comparison results of model variants on BikeNYC and TaxiBJ datasets.

Model	BikeNYC		TaxiBJ	
	RMSE	MAE	RMSE	MAE
M-Resnet+RSE	4.45	1.84	15.82	9.34
3D densenet+M-Resnet	4.05	1.67	15.42	9.16
3D densenet+RSE	4.88	1.98	16.23	9.43
3D densenet+M-Resnet+RSE	3.61	1.53	14.15	8.80
ST-3DDMCRN	3.55	1.49	13.85	8.26

3D densenet+M-Resnet+RSE: This variant does not consider the external factors and directly trains the model for traffic prediction.

ST-3DDMCRN: The proposed model, which includes 3D densenet, M-Resnet, RSE unit, and external factors, unites the advantages of each module to obtain better performance.

Table 8 demonstrates the RMSE results of our proposed model and its variants on the BikeNYC and TaxiBJ datasets. It is observed that directly concatenating the M-Resnet and an RSE unit gains an RMSE of 4.45 on BikeNYC and 15.82 on TaxiBJ. When capturing the low-level spatio-temporal correlation by adding a 3D densenet module, The "3D densenet+M-Resnet+RSE" model can decrease RMSE to 3.61 on BikeNYC and 14.15 on TaxiBJ, with 18.8% and 10.55% relative performance improvement, which indicates the effectiveness of the 3D densenet. Similarly, directly concatenating the 3D densenet and M-Resnet block obtains an RMSE of 4.05 on BikeNYC and 15.42 on TaxiBJ. When obtaining the spatio-temporal heterogeneity by adding the RSE unit, "3D densenet+M-Resnet+RSE" model can decrease RMSE to 3.61 on BikeNYC and 14.15 on TaxiBJ, with 10.86% and 8.23% relative performance improvement, which indicates the effectiveness of the RSE unit. For capturing complex dependence, adding the M-Resnet block to the "3D densenet+RSE" model can improve 26.02% on BikeNYC and 12.65% on TaxiBJ. Considering external conditions, we find that the performance of the prediction

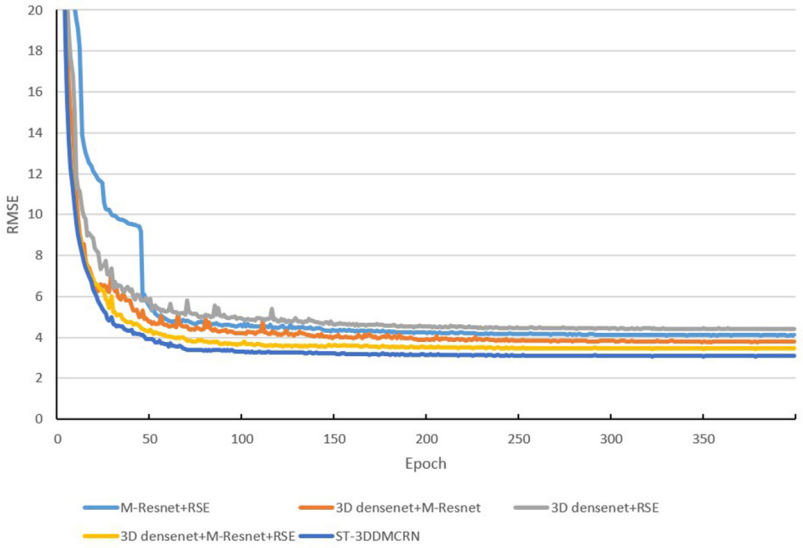


Fig. 16. RMSE of the proposed ST-3DDMCRN model versus different epochs and comparisons with other variants on BikeNYC dataset.

Table 9
Summary of calibrated hyper-parameters on BikeNYC.

ST-Resnet	DeepSTN+	STDN	ST-3Dnet	ST-3DDMCRN
residual number: 4 Channels: 32 Kernel size: (3, 3)	Resplus unit: 2 Channels in Resplus unit: 32 Channels: 64 Kernel size: (1, 1) Pooling rate: 1 Drop out: 0.1	Batch_size = 64 Neighbourhood: 7*7 Short-term LSTM: 7 Long-term LSTM: 3 Shifted attention: 3 dropout in LSTM: 0.5 Kernel size (3, 3) hidden output: 128	Batch_size: 32 3D layer: 3 kernel size: (3, 3, 3) Residual_unit: 4 Resnet_kernel _size: (3, 3)	Batch_size: 32 3D layer in 3D densenet: 3 kernel size: (3, 3, 3) M-Resnet: 2 Covlstm_Resnet_kernel_size (3, 3)

Table 10
Efficiency evaluation on BikeNYC.

Model	Total number of parameters	Time (s) each Epoch	Epochs to converge
ST-ResNet	174,465	6	54
DeepSTN+	32,575,693	52	62
STDN	6,283,522	562	56
ST-3Dnet	543,602	3	169
ST-3DDMCRN	494,884	20	70

has been further improved as well on the BikeNYC and TaxiBJ datasets. Therefore, external information is helpful to the prediction results.

Meanwhile, we visualize the whole training process of each variant on the BikeNYC dataset without early-stopping the mechanism. As shown in Fig. 16, the prediction results of all models have improved as the epoch has increased. In contrast, the ST-3DDMCRN model is still superior to other model variants. It should be noted that when using the 3D densenet block, the training and convergence speed of the model become faster.

5.3.4. Efficiency comparison

This section compares the efficiency of different methods regarding calculation time on the BikeNYC dataset.

For all methods, we consider the parameter settings recommended by the original study and apply grid search to tune the parameters based on the same validation dataset. All these methods are performed on an NVIDIA 2080ti GPU. The best set of hyperparameters is shown as in Table 9. Table 10 gives the total number of parameters and training time for each epoch in the model. It should be noted that: (1) DeepSTN+ utilized a ResPlus unit to capture citywide spatial correlation leading to more parameters than others. (2) STDN exploited 2DCNN to take the grid as its computing unit leading to many parameters. (3) Owing to more Resnet units and "RC" blocks, the ST-3Dnet model has more parameters than the ST-3DDMCRN model. In brief, the whole efficiency of all methods is within the controllable range. Therefore, we should pay more attention to the improvement of model performance.

5.3.5. Visualization and robustness analysis

In this section, we visualize the main core layers of the model and analyse the robustness of the predicted results.

The central core layers of the model contain a 3D densenet layer, M-Resnet layer, RSE layer, and Fusion layer. Fig. 17 displays the visualization result of all layers. The 3D densenet layer consists of three 3D convolution layers. As shown in Figs. 17(a)–17(f), the feature maps have some silhouette features of the traffic flow by 3D convolution, which demonstrates some target features have been captured. After the 3D densenet lay, the feature map seems to imply local spatial characteristics (see Figs. 17(e)–17(f)). Figs. 17(g)–17(n) show the visualization results of the different dilated rates in the M-Resnet layer. It can be viewed that the receptive field of dilated convolution expands with the increase of dilated rate. Compared to the standard Resnet network, the M-Resnet network can capture multiscale spatial dependence from the probably hidden features. We can see that the feature maps in the RSE layer can display the difference of the contributions in space in the closeness, daily-period, and weekly-period modules in Figs. 17(o)–17(q). Fig. 17(r) shows the last fusion layer's predicted result.

In the practical application, as the volume of traffic flow is not uniformly distributed in the whole city, we should pay more

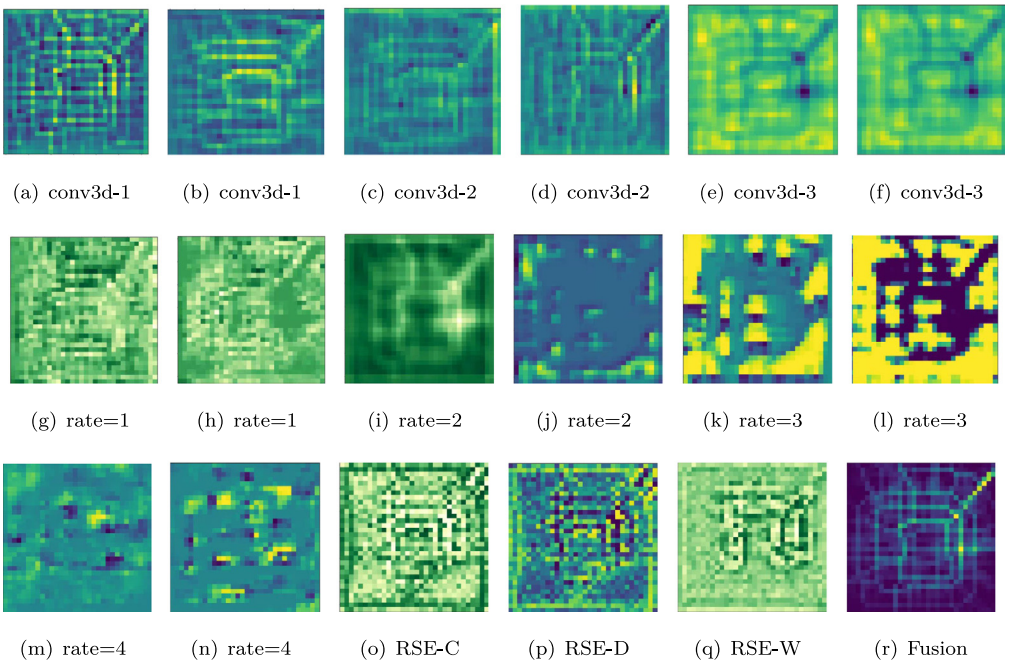


Fig. 17. Feature map visualization of ST-3DDMCRN in core layers on TaxiBJ.

attention to the prediction results of congested areas. To demonstrate the proposed model is worthy of recommendation, we divide traffic flow into three categories (e.g., high, middle, and low capacity) according to different flow volumes. In Fig. 18, taking the BikeNYC data set as an example, we pick up three different flow volume areas to observe the inflow and outflow of the predicted results in a week. It is worth noting that the error between the predicted and the actual value is within an acceptable range, indicating that the proposed model is stable. Accordingly, it is recommended to apply the method proposed in practice.

5.3.6. Multi-step citywide traffic flow prediction

Traffic flow is also foreseeable in multi-step prediction, which is beneficial for traffic control compared to the single-step prediction. Fig. 19 shows the comparison result for four-step prediction. Compared with other algorithms, the ST-3DDMCRN model achieves the best result at each step. Owing to overlooking the multiscale spatial dependence, the multi-step error of ST-3Dnet and ST-LMST3D-Resnet models is larger than that of the ST-3DDMCRN. Notably, the ST-3DDMCRN has performance progress of 12.3% and 18.5% compared with ST-Resnet at the fourth time interval on BikeNYC and TaxiBJ data. Therefore, It is strongly recommended that our model can be used to reference future traffic congestion signs.

5.4. Expand to the passenger demand prediction of the whole city

ST-3DDMCRN model is a general-purpose model which can be used not only for traffic flow prediction but also for passenger demand prediction tasks. Our proposed approach is expanded to predict this section's taxi passenger demand task. The TaxiNYC dataset,[13] which comes from NYCTLC (New York City Taxi and Limousine Commission), is a benchmark for citywide passenger pickup/dropoff demand prediction. A brief description of the data set is shown in Table 11.[14] It mainly includes 132 million NYC taxi trip records in Manhattan in 2014. The dataset divides the Manhattan area into a 15 × 5 grid map based on the longitude and latitude. The timestamp and the geo-coordinates of origin and destination positions are recorded in each grid area every half an hour. Therefore, the passenger demand diagram size in the whole city is 2 × 15 × 5. 2 denotes pickup and dropoff situation. External information contains holidays, temperatures and 17 kinds of weather conditions. The first eleven months' data are chosen as training data, and the last one month is taken as test data.

Table 11
TaxiNYC Dataset.

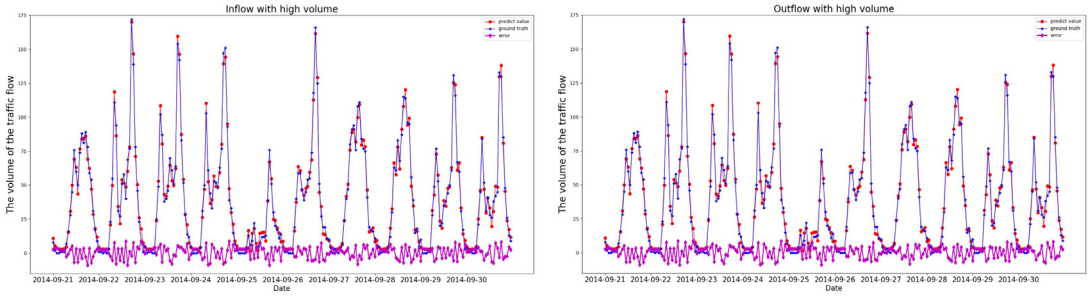
Dataset	TaxiNYC
City	Manhattan
Grid map size	(15, 5)
Time span	1st/1/2014–12th/31/2014
Time interval	half an hour
Available time interval	17520
Holiday	11
Weather condition	17 types
Temperature/°C	20–33
Wind velocity/mph	[0, 22]

We compare the ST-3DDMCRN with the other three deep learning methods based on the best outcomes in the same test dataset and utilize RMSE and MAE as the evaluation benchmark. We use the model with the highest average accuracy of ten validation set. The comparison result is shown in Fig. 20. ST-3DDMCRN model significantly reduces the RMSE and MAE to 15.6 and 11.34, superior to other forecasting methods. In other words, the ST-3DDMCRN model achieves the minimum MAE and RMSE in taxi pickup and dropoff demand tasks. The experimental result demonstrates that our proposed model is general and suitable for taxi passenger demand forecasting.

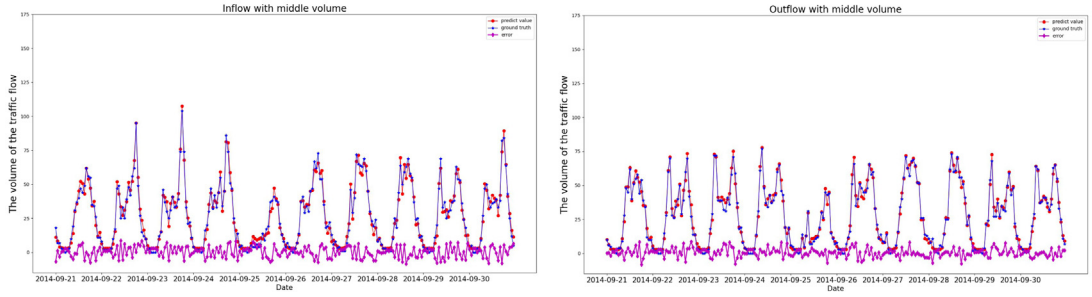
6. Conclusions

Inspired by the 3D CNN-ConvLSTM framework in computer visual spatio-temporal prediction, this article proposes a novel

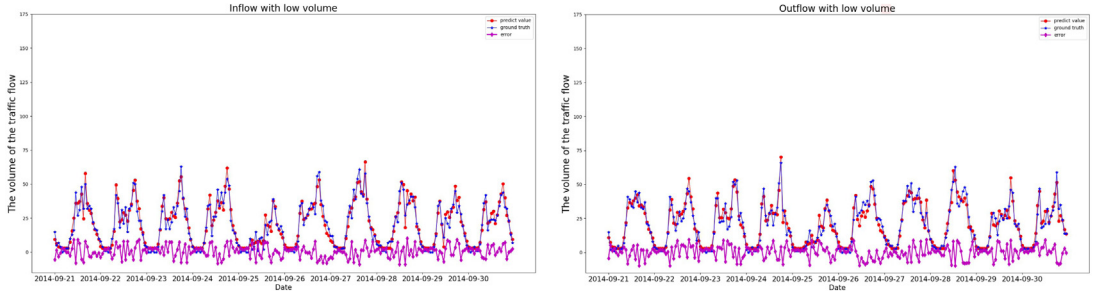
[13] https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page
[14] https://github.com/liulingbo918/ATFM/tree/master/data/TaxiNYC



(a) High Volume

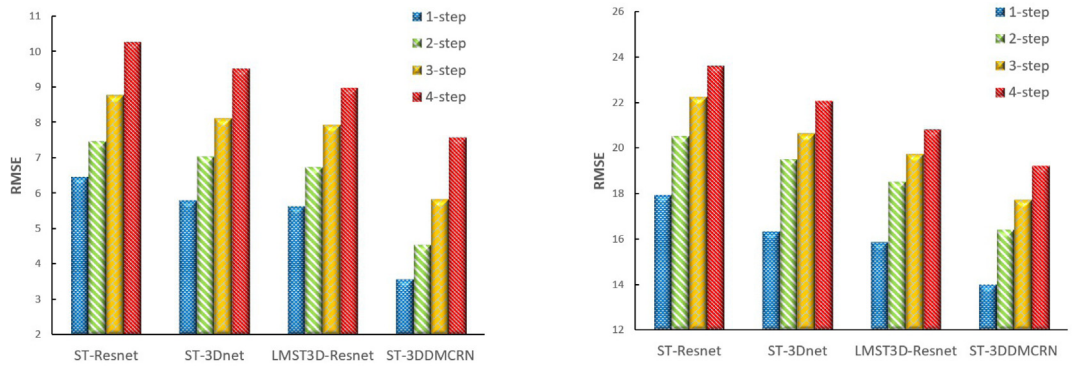


(b) Middle Volume



(c) Low Volume

Fig. 18. Predicted traffic flow with low, middle, high capacity on BikeNYC dataset in one week.



(a) BikeNYC (b) TaxiBJ

Fig. 19. A comparisons four-step prediction on BikeNYC and TaxiBJ.

ST-3DDMCRN network for citywide traffic flow prediction. The ST-3DDMCRN model utilizes a 3D densenet to capture low-level spatio-temporal features and local spatial correlation. Then, the M-Resnet block is developed to simulate multiscale spatial correlation explicitly. A novel RSE unit is designed to explore and calibrate each area's contributions for obtaining the heterogeneity of traffic flow. Experiments have been conducted on two standard benchmarks, demonstrating that ST-3DDMCRN achieves the best performance compared with other baselines. It is noteworthy that the model is also appropriate for other traffic prediction tasks as long as the traffic data can be denoted in the form of spatio-temporal grid data. Therefore, it can be widely used in

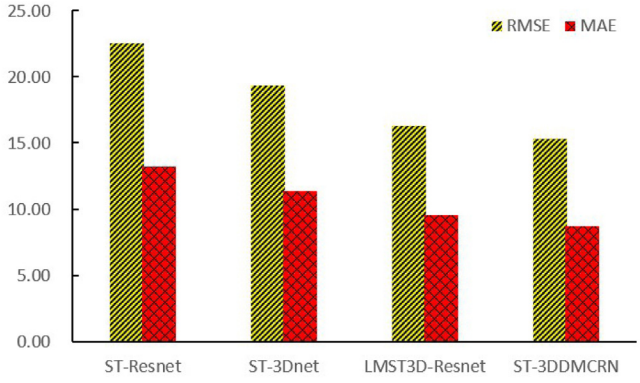


Fig. 20. Quantitative comparison of short-term demand forecasts for citywide passenger traffic.

many traffic forecasting problems in ITS to know traffic status information in advance and to help improve the security and efficiency of ITS.

In future work, we will further study irregular grid regions' traffic flow and focus on regional function information (Land data, MetroCard data, Bus card data, Point of Interest (POI) data, etc.) to further improve prediction accuracy. Meanwhile, we could simulate such traffic systems as graph structure to learn spatio-temporal features by Graph Convolutional Network. https://github.com/346644054/ST-3DDMCRN will be officially released, including data sets and code, if this article is accepted.

CRediT authorship contribution statement

Rui He: Conceptualization, Methodology, Writing – original draft, Writing – review & editing, Software. **Yanbing Liu:** Formal analysis, Writing – review & editing, Funding acquisition. **Yunpeng Xiao:** Supervision. **Xingyu Lu:** Writing – review & editing. **Song Zhang:** Validation, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grants 61772098

References

[1] Yu Zheng, Licia Capra, Ouri Wolfson, Hai Yang, Urban computing: Concepts, methodologies, and applications, ACM Trans. Intell. Syst. Technol. 5 (3) (2014) 38:1–38:55.

[2] Xuan Song, Haoran Zhang, Rajendra Akerkar, Huawei Huang, Song Guo, Lei Zhong, Yusheng Ji, Andreas L. Opdahl, Hemant Purohit, André Skupin, Akshay Pottathil, Aron Culotta, Big data and emergency management: Concepts, methodologies, and applications, IEEE Trans. Big Data 8 (2) (2022) 397–419.

[3] Matthew Veres, Medhat Moussa, Deep learning for intelligent transportation systems: A survey of emerging trends, IEEE Trans. Intell. Transp. Syst. 21 (8) (2020) 3152–3168.

[4] Junbo Zhang, Yu Zheng, Dekang Qi, Ruiyuan Li, Xiuwen Yi, DNN-based prediction model for spatio-temporal data, in: Siva Ravada, Mohammed Eunus Ali, Shawn D. Newsam, Matthias Renz, Goce Trajcevski (Eds.), Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, GIS 2016, Burlingame, California, USA, October 31 – November 3, 2016, ACM, 2016, pp. 92:1–92:4.

[5] Junbo Zhang, Yu Zheng, Dekang Qi, Deep spatio-temporal residual networks for citywide crowd flows prediction, in: Satinder P. Singh, Shaul Markovitch (Eds.), Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4–9, 2017, San Francisco, California, USA, AAAI Press, 2017, pp. 1655–1661.

[6] Huaxiu Yao, Fei Wu, Jintao Ke, Xianfeng Tang, Yitian Jia, Siyu Lu, Pinghua Gong, Jieping Ye, Zhenhui Li, Deep multi-view spatial-temporal network for taxi demand prediction, in: Sheila A. McIlraith, Kilian Q. Weinberger (Eds.), Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th Innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2–7, 2018, AAAI Press, 2018, pp. 2588–2595.

[7] Fazlollah Soleymani, Ali Akgül, European option valuation under the bates PIDE in finance: A numerical implementation of the Gaussian scheme, Discrete Contin. Dyn. Syst. Ser. S 13 (3) (2020) 889.

[8] Yasin Fadaei, Zareen A. Khan, Ali Akgül, A greedy algorithm for partition of unity collocation method in pricing American options, Math. Methods Appl. Sci. 42 (16) (2019) 5595–5606.

[9] Fazlollah Soleymani, Ali Akgül, Improved numerical solution of multi-asset option pricing problem: A localized RBF-FD approach, Chaos Solitons Fractals 119 (2019) 298–309.

[10] Huaxiu Yao, Xianfeng Tang, Hua Wei, Guanjie Zheng, Zhenhui Li, Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction, in: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, the Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, the Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 – February 1, 2019, AAAI Press, 2019, pp. 5668–5675.

[11] Cen Chen, Kenli Li, Sin G. Teo, Guizi Chen, Xiaofeng Zou, Xulei Yang, Ramaseshan C. Vijay, Jiashi Feng, Zeng Zeng, Exploiting spatio-temporal correlations with multiple 3D convolutional neural networks for citywide vehicle flow prediction, in: IEEE International Conference on Data Mining, ICDM 2018, Singapore, November 17–20, 2018, IEEE Computer Society, 2018, pp. 893–898.

[12] Shengnan Guo, Youfang Lin, Shijie Li, Zhaoming Chen, Huaiyu Wan, Deep spatial-temporal 3D convolutional neural networks for traffic data forecasting, IEEE Trans. Intell. Transp. Syst. 20 (10) (2019) 3913–3926.

[13] Yibi Chen, Xiaofeng Zou, Kenli Li, Keqin Li, Xulei Yang, Cen Chen, Multiple local 3D CNNs for region-based prediction in smart cities, Inform. Sci. 542 (2021) 476–491.

[14] Liang Zhang, Guangming Zhu, Peiyi Shen, Juan Song, Learning spatiotemporal features using 3DCNN and convolutional LSTM for gesture recognition, in: 2017 IEEE International Conference on Computer Vision Workshops, ICCV Workshops 2017, Venice, Italy, October 22–29, 2017, IEEE Computer Society, 2017, pp. 3120–3128.

[15] Guangming Zhu, Liang Zhang, Peiyi Shen, Juan Song, Syed Afaq Ali Shah, Mohammed Bennamoun, Continuous gesture segmentation and recognition using 3DCNN and convolutional LSTM, IEEE Trans. Multimed. 21 (4) (2019) 1011–1021.

[16] Tian Wang, Jiakun Li, Mengyi Zhang, Aichun Zhu, Hichem Snoussi, Chang Choi, An enhanced 3DCNN-ConvLSTM for spatiotemporal multimedia data analysis, Concurr. Comput. Prac. Exper. 33 (2) (2021).

[17] Kui-Lin Li, Chun-Jie Zhai, Jian-Min Xu, Short-term traffic flow prediction using a methodology based on ARIMA and RBF-ANN, in: 2017 Chinese Automation Congress, CAC, 2017, pp. 2804–2807.

[18] Zhiyong Cui, Kristian Henrickson, Ruimin Ke, Yinhai Wang, Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting, IEEE Trans. Intell. Transp. Syst. 21 (11) (2020) 4883–4894.

[19] Hongjie Liu, Hongzhe Xu, Yu Yan, Zaishang Cai, Tianxu Sun, Wen Li, Bus arrival time prediction based on LSTM and spatial-temporal feature vector, IEEE Access 8 (2020) 11917–11929.

[20] Wenchao Tian, Wenju Li, Multi-mode spatial-temporal convolution network for traffic flow forecasting, in: 2021 2nd International Conference on Big Data and Informatization Education, ICBDIE, 2021, pp. 278–281.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27–30, 2016, IEEE Computer Society, 2016, pp. 770–778.

[22] Ziqian Lin, Jie Feng, Ziyang Lu, Yong Li, Depeng Jin, DeepSTN+: Context-aware spatial-temporal neural network for crowd flow prediction in Metropolis, in: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, the Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, the Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 – February 1, 2019, AAAI Press, 2019, pp. 1020–1027.

[23] S. Du, T. Li, X. Gong, S.J. Horng, A hybrid method for traffic flow forecasting using multimodal deep learning, Int. J. Comput. Intell. Syst. 13 (1) (2018).

[24] Renhe Jiang, Zekun Cai, Zhaonan Wang, Chuang Yang, Zipei Fan, Quanjun Chen, Kota Tsubouchi, Xuan Song, Ryosuke Shibasaki, DeepCrowd: A deep model for large-scale citywide crowd density and flow prediction, IEEE Trans. Knowl. Data Eng. (2021).
[25] J. Haworth, T. CHENG, Spatio-temporal autocorrelations of networks and their implications for space-time modelling, 2011.
[26] Du Tran, Lubomir D. Bourdev, Rob Fergus, Lorenzo Torresani, Manohar Paluri, Learning spatiotemporal features with 3D convolutional networks, in: 2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7–13, 2015, IEEE Computer Society, 2015, pp. 4489–4497.
[27] Xiaopeng Ji, Qingsong Zhao, Jun Cheng, Chenfei Ma, Exploiting spatio-temporal representation for 3D human action recognition from depth map sequences, Knowl. Based Syst. 227 (2021) 107040.
[28] Jinglong Du, Lulu Wang, Ali Gholipour, Zhongshi He, Yuanyuan Jia, Accelerated super-resolution MR image reconstruction via a 3D densely connected deep convolutional neural network, in: Huiru Jane Zheng, Zoraida Callejas, David Griol, Haiying Wang, Xiaohua Hu, Harald H.H.W. Schmidt, Jan Baumbach, Julie Dickerson, Le Zhang (Eds.), IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2018, Madrid, Spain, December 3–6, 2018, IEEE Computer Society, 2018, pp. 349–355.
[29] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, Yun Fu, Residual dense network for image restoration, IEEE Trans. Pattern Anal. Mach. Intell. 43 (7) (2021) 2480–2495.
[30] David Harris, Sarah L. Harris, Digital Design and Computer Architecture, Morgan Kaufmann, 2010.
[31] Haifeng Zheng, Feng Lin, Xinxin Feng, Youjia Chen, A hybrid deep learning model with attention-based conv-LSTM networks for short-term traffic flow prediction, IEEE Trans. Intell. Transp. Syst. 22 (11) (2021) 6910–6920.
[32] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, Enhua Wu, Squeeze-and-excitation networks, IEEE Trans. Pattern Anal. Mach. Intell. 42 (8) (2020) 2011–2023.
[33] Diederik P. Kingma, Jimmy Ba, Adam: A method for stochastic optimization, in: Yoshua Bengio, Yann LeCun (Eds.), 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings, 2015.
[34] Xavier Glorot, Yoshua Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Yee Whye Teh, D. Mike Titterington (Eds.), Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2010, Chia Laguna Resort, Sardinia, Italy, May 13–15, 2010, in: JMLR Proceedings, vol. 9, JMLR.org, 2010, pp. 249–256.