Information Sciences 542 (2021) 476–491

Contents lists available at ScienceDirect

Information Sciences

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

Multiple local 3D CNNs for region-based prediction in smart cities

Yibi Chen [a,1], Xiaofeng Zou [a,1], Kenli Li [a,*], Keqin Li [a,b], Xulei Yang [c], Cen Chen [a,d]

[a] College of Computer Science and Electronic Engineering, Hunan University, China
[b] Department of Computer Science, State University of New York, New Paltz, NY 12561, USA
[c] YITU Technology, Singapore
[d] Institute for Infocomm Research, Agency for Science, Technology and Research (A*STAR), Singapore

ARTICLE INFO

Article history:
Received 23 December 2019
Received in revised form 18 May 2020
Accepted 6 June 2020
Available online 25 June 2020

Keywords:
Electricity prediction
Local 3D CNNs
Residual neural network
Spatial-temporal features
Traffic vehicle prediction

ABSTRACT

In smart cities, region-based prediction (e.g. traffic flow and electricity flow) is of great importance to city management and public safety, and it remains a daunting challenge that involves complicated spatial-temporal-related factors such as weather, holidays, events, etc. Region-based forecasting aims to predict the future situation for regions in a city based on historical data. In the existing literature, the state-of-the-art method solve region-based problems with long short-term memory (LSTM) algorithms that extract the temporal view and local convolutional neural network (CNN) algorithms that extract the spatial view (local spatial correlation via local CNN). In this paper, we propose a deep learning-based method for region-based prediction for smart cities. First, we divide the cities into regions based on the space dimension and model the situation of the cities in 3D volumes. Based on the constructed 3D volumes, we design a model called multiple local 3D CNN spatial-temporal residual networks (LMST3D-ResNet) for region-based prediction in smart cities. LMST3D-ResNet can extract multiple temporal dependencies (including trend, period and closeness) for local regions and then predict the future citywide activities according to the learned multiple spatial-temporal features. LMST3D-ResNet can also combine the spatial-temporal features with external factors. LMST3D-ResNet includes 3D CNNs and ResNet mechanisms for processing spatial-temporal information. In particular, 3D CNNs have the ability to model 3-dimensional information due to 3D convolution and 3D pooling operations, while ResNet enables the connection of the convolutional neural network across layers to obtain a deeper network structure. Specifically, in our proposed model, a novel region-based information extraction mechanism and an end-to-end multiple spatial-temporal dependency learning structure are designed for local regions. Extensive experimental results on two datasets, i.e., MLElectricity and BJTaxi demonstrate the superior performance of our proposed method over the exisiting state-of-the-art methods.

* Corresponding author.
E-mail address: lkl@hnu.edu.cn (K. Li).
[1] These authors are with equal contributions.

https://doi.org/10.1016/j.ins.2020.06.026
0020-0255/

1. Introduction

With the massive use of computing devices and the embedding of smart technologies [1,2] in these devices, cities have become smarter, more conscious, and faster through computing devices [3] and smart technologies [4,5]. Accurate prediction for regions in a city is crucial for smart cities. An increasing number of researchers attempt to leverage deep learning techniques to forecast situations for city regions [6]. Region-based predictions can obtain a more accurate future situation because each local region in the city has different regional characteristics and is affected by its nearby and more distant regions. The situational change in local regions is a dynamic and real-time process; thus, it has great challenges. For example, region-based prediction of traffic volume can help the transportation department better manage traffic to cope with various situations (i.e., traffic accidents). Furthermore, the transportation department can promptly inform vehicles in other regions not to enter the region to avoid more serious congestion. We study region-based prediction and establish local spatial correlations with other regions. The spatial-temporal correlation among multiple local regions of the city aims to predict the activity of the regions in the future based on historical information.

Regarding the region-based prediction of smart cities, electricity usage prediction, and traffic vehicle prediction are important evaluation indicators. For instance, on August 14, 2003, a quarter of the US suffered an electricity outage. David Rosenberg, the chief economist at Merrill Lynch, estimated that the entire economic loss was between 25 billion and 30 billion. The 2018 Shanghai Autumn and Winter Road Traffic Safety Work Conference revealed that there were 809 road traffic accidents in Shanghai, causing 619 deaths and 385 injuries. In this paper, we take the electricity usage and traffic flow prediction as examples to show the effectiveness of our proposed model. The region-based predictions are affected by the following factors.

Factor 1: Local regional spatial correlations The spatial correlation of local regions is affected by spatially nearby regions. For example, residential and industrial areas may cause large changes when people from living areas commute to industrial areas for work. Furthermore, residential and commercial areas may cause substantial changes when people from living areas travel to commercial areas for shopping during the holidays. Therefore, the dependencies of multiple local regions are extremely important for region-based prediction.

Factor 2: Temporal correlations For the local region i, j, different temporal intervals (i.e., recent, nearby and long) affect the state of region-based prediction. For instance, the condition of electricity and traffic at 7 pm will affect the next temporal intervals in the region.

Factor 3: External module factors The external module factors include accidents, events, and weather conditions. The state of the local region i, j is related to external module factors. For example, in most cities, the changes in traffic vehicles and electricity usage are large on Fridays from 6 pm to 7 pm.

Many of the techniques in the literature are applied to region-based predictions. The autoregressive integrated moving average (ARIMA) and its variants have been widely applied to solve traffic prediction for time-series [7–9]. Some research considers spatial correlation [10,11] and increases in external factors [12,13] (e.g., events, climate and weather); these methods improve the prediction accuracy, but they are only for linear features. With the rapid development of deep learning, neural networks model the relationship of multidimensional nonlinear features (e.g., image segmentation, object detection, and natural language processing) [14–16]. Some research [17,6] has recommended treating the region in the city as an image. Given a set of historical region images, the model predicts the region image change for the next timestamp. Convolutional neural network (CNN) [18] are used to simulate complex spatial correlations. [19] has recommended using long short-term memory (LSTM) [20] networks to predict loop sensor readings. They showed that the proposed LSTM model can model complex sequential features. These methods show superior performance compared to previous methods based on traditional time-series prediction methods. However, they did not consider correlation between both the spatial and temporal sequential relationships. The weak correlation of local regions actually deteriorates the performance of prediction in a target region.

ST-ResNet [6] is based on a convolutional residual network to simulate the proximity and long-range spatial correlation between any two regions in a city. STDN [21] captures the temporal and spatial features of traffic data through LSTM and local 2D CNNs. However, they do not consider or do not fully use the low-level spatial-temporal correlation features and dependency features among local regions. Therefore, this can cause an ineffective extraction of some features that would otherwise enable achieving better accuracy.

To address these challenges, we propose LMST3D-ResNet for the spatial-temporal correlation among multiple local regions of the city to predict the future situation for regions in a city based on historical data. The main motivation for LMST3D-ResNet is the spatial-temporal correlation features in multiple local regions can be extracted and learned simultaneously to achieve historical data mining from low-level to high-level layers. The proposed LMST3D-ResNet establishes a region-based prediction model that considers the three types of factors as described above. The 3D CNNs process 3-dimensional information through the 3D convolution and the 3D pooling layer, which can effectively learn the spatial-temporal features of local regions. Furthermore, ResNet enables the connection of the convolutional neural network across layers to achieve a deeper network structure. LMST3D-ResNet can extract multiple temporal dependencies (including trend, period and closeness) for local regions. It dynamically aggregates the three network outputs (i.e., trend, period and closeness), and different weights are assigned to different branches. Our method is validated on the MLElectricity and BJTaxi datasets. The MLElectricity dataset contains one month of continuous data with a daily interval of ten minutes. The BJTaxi

datasets consist of four time intervals of continuous data with a daily interval of a half hour. We compare our proposed method with state-of-the-art methods and demonstrate its superior performance.

We summarize the main contributions of this paper as follows:

- We propose a novel method called LMST3D-ResNet for learning the spatial-temporal features of citywide local regions by local 3D CNNs. It considers the local regional spatial correlation, temporal correlation, and external module factors. LMST3D-ResNet can combine the external module factors with the spatial-temporal features of multiple local 3D CNNs outputs.
- LMST3D-ResNet is an end-to-end structure. It dynamically aggregates the three network outputs (i.e., trend, period and closeness), and different weights are assigned to different branches. The aggregation is combined with the output of the external factors.
- Our approach is appropriate for different types of datasets and can conduct different kinds of region-based predictions to obtain different evaluation factors for citywide areas. LMST3D-ResNet demonstrates better flexibility in region-based prediction.
- We validate our approach on the MLElectricity and BJTaxi datasets. The experimental results show that our proposed LMST3D-ResNet is superior to the state-of-the-art methods.

2. Related work

With the development of region-based predictions, increasing literature has been published on different prediction indicators, such as electricity, air pollution, and traffic vehicles. Different predicting approaches have been applied to smart cities [22].

For time-series prediction, different linear and nonlinear models are constructed. The autoregressive model (AR), uses its own process of regression variables. That is, the linear combination of random variables at some time in the previous period is used to describe the linear regression model of random variables at some later time. The moving average model (MA) increases the smoothing fluctuation effect by increasing the number of periods of the moving average method. Historical average (HA) is an estimate of the inflow and outflow of the region based on the average of the previous relative time intervals in the citywide regions. The autoregressive integrated moving average (ARIMA) [23] model is widely applied [7–9] to solve traffic prediction for time-series. Vector autoregressive (VAR) [24,25] models are commonly used to predict interconnected time series and to analyze the dynamic effects of random changes on variable systems. The LSTM model [20] extracts long-term and short-term-related information from time trajectories. LGnet [26] proposes to solve the multivariate time series with missing values through the memory network and adversarial training to model the global temporal distribution. Soto et al. [27] propose different fuzzy aggregation models to perdict multiple time series. These models can provide better correlation over a continuous time series [28]. However, they are unable to capture most of the important spatial features within the city regions.

For spatial prediction, some works explore various techniques to simulate spatial interactions. LSM-RN [10] predicts citywide region traffic by using matrix decomposition on the road network to capture spatial correlations between road junction regions. The existing latent spaces are adjusted and updated by the sensor data and the data are used for training and real-time prediction. LinUOTD [11] uses a simple model structure to avoid the repetitive design of the model and a linear regression model for regularization of spatial-temporal features. XGBoost [29] is a method of building and extending trees based on approximate tree learning algorithms and sparsity-aware algorithms, which can extend more unknown resources with fewer resources available. Sugiyama et al. [30] have solved the problem of trajectory cost by a weight propagation mechanism and simplified the problem into a kernel ridge regression problem. Zheng et al. [31] have proposed normalizing the prediction differences between nearby locations and time points to obtain near spatial and temporal correlation. In addition, [12,13] have further explored the utility of external environmental data, such as events, climate, and weather. However, these methods fail to simulate complex nonlinear spatial-temporal relationships.

Recently, the success of deep learning has prompted researchers to apply deep learning techniques to citywide region prediction problems. For example, Ma et al. [32] use CNN on traffic images automatically by extracting spatial-temporal features of network traffic. DeepSD [33] can be regarded as a kind of multilayer perceptron (MLP)-extended model that addresses the supply and demand mode and uses external environment information (i.e., weather and traffic information.) In ST-ResNet [6], the residual network is implemented on traffic flow images. Yu et al. [19] suggested applying an LSTM network and an autoencoder to capture the sequential correlation between predicted traffic under extreme conditions for both normal and peak hours. STDN [21] is proposed as a multiview spatial-temporal network that combines local 2D CNNs, LSTM and an attention mechanism to predict traffic conditions. MST3D [34] is proposed implementing multiple 3D CNNs to make full use of low-level spatial-temporal correlation features. In these studies, they do not consider or do not fully use some features in the local region, such as low-level spatial-temporal correlation features and dependency features among local regions.

In summary, our method is very different from the above literature. In our proposed model, a novel region-based information extraction mechanism and the spatial-temporal correlations among multiple local regions of the city are designed. The spatial-temporal correlation features in multiple local regions can be extracted and learned simultaneously to achieve historical data mining from low-level to high-level layers.

3. Preliminaries

In this section, we define some concepts. Based on these concepts, we propose a region-based prediction problem. According to this problem, some definitions are provided by taking BJTaxi and MLElectricity datasets as examples. The details are as follows.

Definition 1. (City Local Region Segmentation): According to previous studies [6,10,32], based on longitude and latitude, the city is divided into an $I \times J$ grid map size, where the grid map represents the region. We define each local region in the grid map, expressed as nonoverlapping pairs $(i,j) \in (I,J)$. i and j represent the ith row and the jth column, respectively, of each local region.

Problem 1 (City Region-based Prediction): Obtain a set of historical citywide regional spatial-temporal data with a time interval of $T = 1, 2, \ldots, t-1$, where the citywide region is composed of multiple local regions. The problem of smart city region-based prediction aims to predict region situations of the next time interval t by obtaining the historical spatial-temporal feature of each local region.

Definition 2. (City Traffic Flow): According to previous studies [6,34], we collect traffic records for time interval t, denoted as P. We take the inflow and output of the BJTaxi dataset as an example in Figure 1. At time interval t, the traffic inflow and outflow of each local region (i,j) are defined as Eq. 1 and Eq. 2, respectively.

$$x_t^{in,i,j} = \sum_{Tr \in \mathrm{P}} |\{\lambda > 1 | g_{\lambda-1} \notin (i,j) \wedge g_\lambda \in (i,j)\}| \tag{1}$$

$$x_t^{out,i,j} = \sum_{Tr \in \mathrm{P}} |\{\lambda \geqslant 1 | g_\lambda \in (i,j) \wedge g_{\lambda+1} \notin (i,j)\}| \tag{2}$$

where $Tr : g_1 \rightarrow g_2 \rightarrow, \ldots, \rightarrow g_{|Tr|}$ is a trajectory in P, λ is the number of local regions, and g_λ is the geospatial coordinate, $g_\lambda \in (i,j)$ represents the point g_λ that lies within grid (i,j), and vice versa.

Definition 3. (City Electricity Usage): The difference between the citywide electricity flow and the definition of city traffic flow is that it only considers inflows and does not need to consider outflows. The inflow of the electricity at the time interval t is defined as Eq. 1

4. The proposed LMST3D-ResNet framework

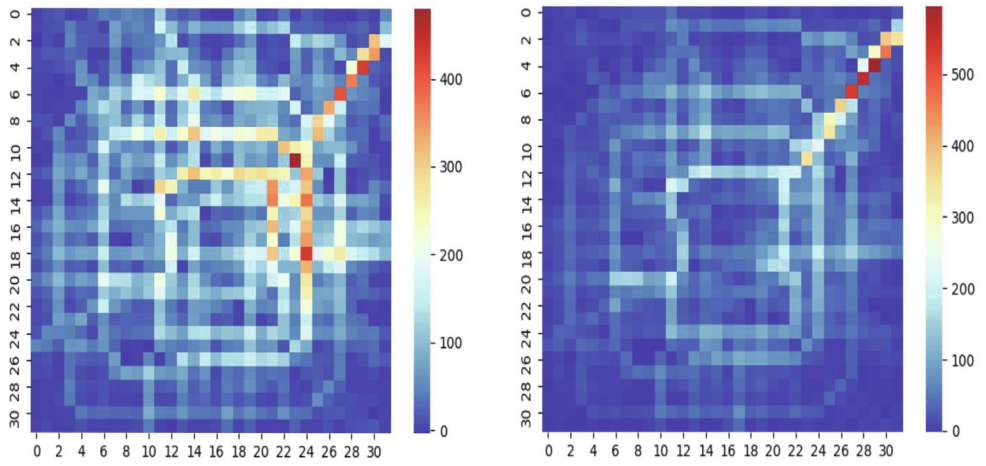
In this section, we present a detailed description of our proposed LMST3D-ResNet framework using a region-based prediction method to predict the future situation for regions in a city. The architecture of our proposed method is shown in Figure 2. The model we propose has four aspects: local 3D CNNs, ResNet, weighted feature fusion and external module fusion.

We learn the spatial-temporal correlation features in multiple local regions through LMST3D-ResNet. The feature of the spatial view is that the surrounding neighborhood of each local range is centered on $R_{i,j}$. In Figure 2, the red line represents the inflow from the spatially nearby regions to the current region; the black line represents the outflow from the current region to spatially nearby regions. The features of the temporal view include different trend, period and closeness correlations, represented by pink, blue and yellow, respectively. The 3D convolutional layer and 3D pooling layer is represented by light blue cubes and light green cubes, respectively. In each branch, the attribute features described above are learned by a single 3D CNNs. LMST3D-ResNet dynamically aggregates the three network outputs (i.e., trend, period and closeness), and different weights are assigned to different branches. External modules include climatic conditions, event information, and other factors. We manually extract information from these modules. Then, we pass this information to the two fully connected layers. Finally, the external module information fusion and the spatial-temporal features are combined to calculate the loss.

4.1. Local 3D CNNs

We first explain why the spatial-temporal correlation among multiple local regions is predicted by multiple local 3D CNNs to predict the future situation for regions in a city based on historical data. Then, we analyze the impact of multiple temporal correlations on region-based predicting and proposed a process for modeling multiple temporal correlations in our approach.

The idea of local 2D CNNs is to solve the weak correlation between the current region and the neighboring regions, which mainly represents the spatial view. Due to the limitations of 2D CNNs, temporal features can only be learned using another method (e.g., LSTM). However, this method can only establish temporal connections on the high-level layer spatial-temporal features, while low-level layer spatial-temporal features are not fully utilized. The 3D CNNs process 3-dimensional information through the 3D convolution and the 3D pooling layer, which can effectively learn the spatial-temporal features. The 3D convolution equation is as follows:



(a) $Inflow$ (b) $Outflow$

Fig. 1. Inflow and outflow in every region of Beijing.

$$u_{ij}^{\beta}(x,y,z) = \sum_{m,n,l} V_i^{\beta-1}(x-m, y-n, z-l) W_{ij}^{\beta}(m,n,l), \tag{3}$$

The 3D feature equation in the βth layer taking the trend branch as an example is:

$$V_j^{\beta} = f\left(\sum_i u_{ij}^{\beta} + b_j^{\beta}\right), \tag{4}$$

where f is the rectified linear unit (ReLU) function and b is the bias term in the feature map.

With 3D CNNs, although the low-level features are better utilized, the weak correlation between regions still deteriorates performance. To solve this problem, we propose that the local 3D CNNs method considers both the temporal limit of the local 2D CNNs and the limitation of the local spatial region correlation of 3D CNNs. It improves the spatial correlation by establishing the spatial-temporal dependency of multiple local regions.

4.1.1. Local spatial correlation

We regard the surrounding region as an $L \times L$ image with $R_{i,j}$ in the region as the center point. $R_{i,j}$ is expressed as a channel of demand values and controls the spatial granularity by L size. For the boundaries of the local region, we use zero-padding measures. The region image tensor is denoted as $Y_{t-1}^{i,j} \in R^{h \times L \times L \times p}$, for time interval t, each location i,j, time segments h and image channels p. The local 3D CNNs convert $Y_{t-1}^{i,j}$ to $Y_{t-1}^{i,j,0}$ as input and passes it to the N convolutional layers. Taking the nth layer as an example, the transformation is defined as follows:

$$Y_{t-1}^{i,j,n} = f\left(Y_{t-1}^{i,j,n-1} * W_{t-1}^{n} + b_{t-1}^{n}\right), \tag{5}$$

where $*$ represents the 3D convolution operation, f is the activation function that represents $f(c) = max(0,c)$, and W_t^n and b_t^n represent the nth layer of convolution parameters. The two parameters $W_{t-1}^{1,\ldots,N}$ and $b_{t-1}^{1,\ldots,N}$ are shared throughout all regions $i,j \in I,J$.

4.1.2. Temporal correlation

It can be clearly seen from the above spatial correlation that the temporal correlation has a significant impact on region-based prediction. We consider temporal correlation features in three categories, including closeness, period, and trend. In Figures 3 and 4, we show the temporal correlations on the MLElectricity and BJTaxi datasets, respectively.

3D temporal closeness is modeled on recent time intervals of local region images of 2 channels. The recent fragment can be denoted as $\left[Y_{t-l_c}^{i,j}, Y_{t-(l_c-1)}^{i,j}, \ldots, Y_{t-1}^{i,j}\right]$, where l_c is the length of the recent fragment. Its 3D form is expressed as $V_c \in R^{l_c \times i \times j \times p}$.

Similar to the method for handling the period and trend features, l_d is the time interval extracted from the period fragment, where d is the period span. The dependent sequence daily period can be denoted as $\left[Y_{t-l_d \times d}^{i,j}, Y_{t-(l_d-1)\times d}^{i,j}, \ldots, Y_{t-1}^{i,j}\right]$. The

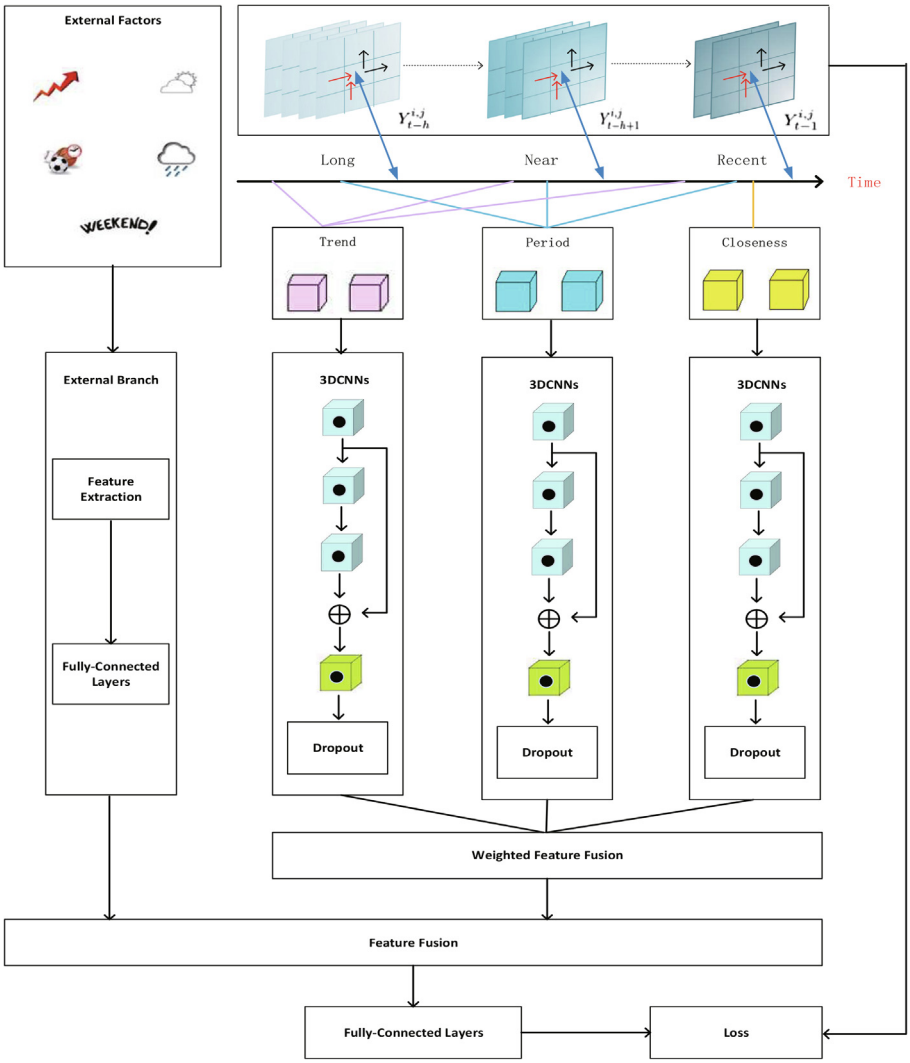
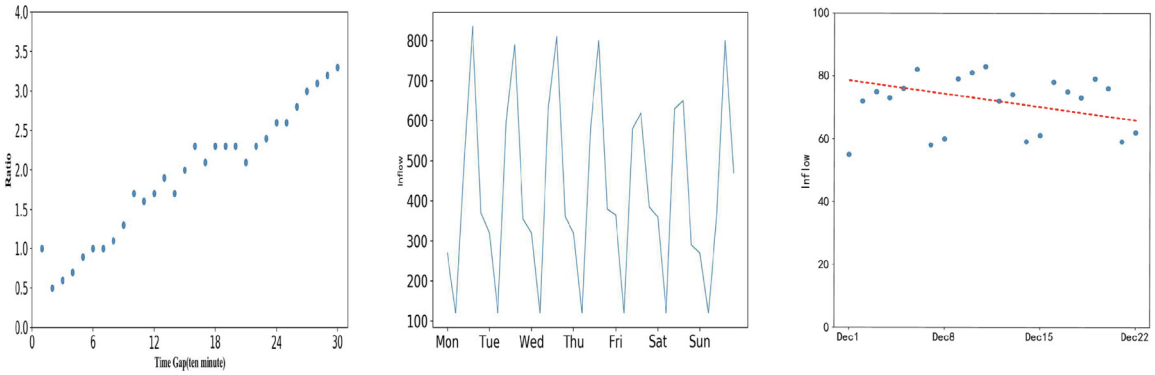
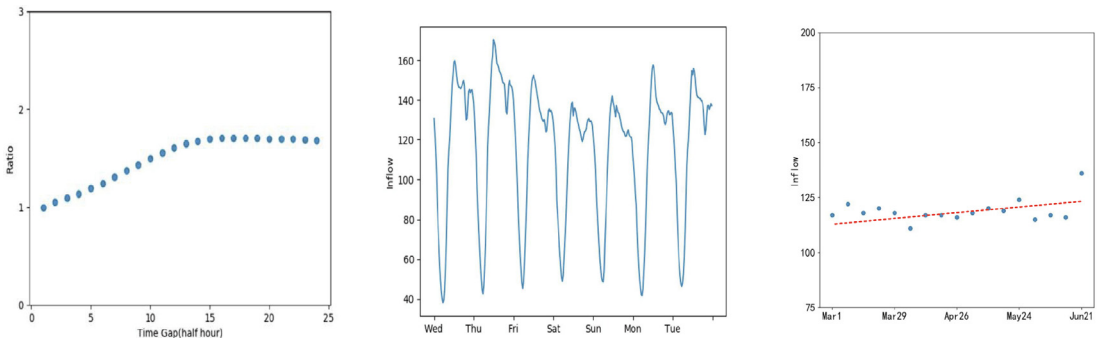


Fig. 2. Architecture of LMST3D-ResNet.



(a) *closeness* (b) *period* (c) *trend*

Fig. 3. Temporal correlations on the MLElectricity dataset.



(a) *closeness* (b) *period* (c) *trend*

Fig. 4. Temporal correlations on the BJTaxi dataset.

sequence 3D form is $V_d \in R^{l_d \times i \times j \times p}$. The trend feature is defined as $\left[Y^{i,j}_{t-l_q \times q}, Y^{i,j}_{t-(l_q-1)\times q}, \ldots, Y^{i,j}_{t-1}\right]$. l_q is the length of the trend correlation sequence and q is the trend span. The input 3D form is expressed as $V_s \in R^{l_q \times i \times j \times p}$.

4.2. Residual network

Convolutional neural network (CNN) are capable of extracting different features based on different network layers. Different features can contribute different information. We assume that the size of the input citywide regions is 8×16, and the kernel of the 3D convolution is $3 \times 3 \times 3$. Modeling the spatial-temporal correlations across multiple local regions of the city involves a relatively large number of convolutional layers. To use the CNN to extract feature information more fully from predicted data of local regions, we increase the depth of the network as much as possible. However, gradient diffusions or gradient explosions can easily occur if the network is arbitrarily deepened.

Therefore, the proposed deep residual learning enables the convolutional neural network to be connected across layers to have a deeper network structure. This structure produces state-of-the-art results in the fields of image classification, object detection, and image segmentation. According to the above discussion, the idea of the deep residual network [35] is added to our model, shown in Figure 5. In our LMST3D-ResNet, the 3D residual network is defined as follows:

$$H = x + F(x), \tag{6}$$

where x is the input to the 3D residual network, $F(x)$ is the 3D residual mapping, and H is the output of the 3D residual network.

By adding a residual structure, the CNN can extract more features from the input data, especially for learning the spatial-temporal features in multiple local regions of the city. Additionally, it is a better approach to adjust the structure of the model to output better results.

4.3. Weighted feature fusion

As discussed in Section 4.1.2, the local region is affected by multiple temporal correlations. Taking the BJTaxi dataset as an example, for city region traffic vehicles, traffic conditions are transmitted through observation sensors. Traffic vehicles change relatively little from 5 am to 6 am. However, the change from 6 pm to 7 pm is relatively large. Trend, period and closeness branches have different effects for local regions because three branches extract different spatial-temporal features. In each branch, the attribute features described above are learned by a 3D CNNs. LMST3D-ResNet dynamically aggregates the three branches outputs (i.e., trend, period and closeness), and different weights are assigned to different branches; the equation is as follows:

$$V_{fusion} = W_c \otimes V_c + W_s \otimes V_s + W_d \otimes V_d \tag{7}$$

where V_{fusion} denotes the fused features in the local region; $\otimes$ is the Hadamard product (i.e., elementwise multiplication for tensors); V_c, V_d, V_s are the features extracted by *closeness*, *period* and *trend* branches respectively; W_c, W_d, W_s are the learnable parameters that adjust the degrees affected by different branches. Then, the fused features in the local region are flattened into a vector feature called V_{m3d}.

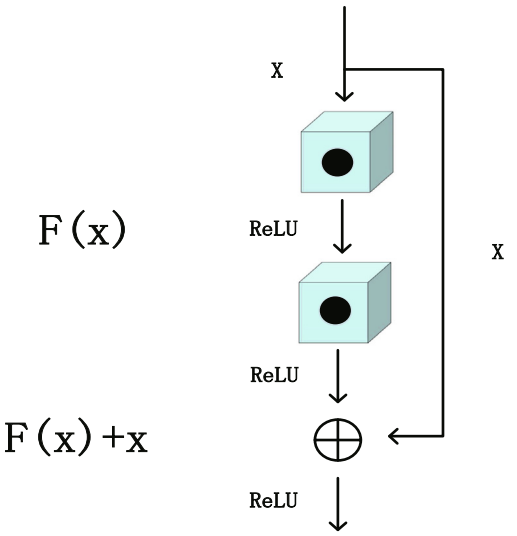


Fig. 5. Flowchart of a 3D convolution residual network. We add a residual mapping to the two 3D convolutional layers.

4.4. External module fusion

As is known, external module factors have a greater impact on the region-based prediction. Through our observations, the number of traffic vehicles in bad weather conditions is significantly reduced compared to normal weather conditions, while electricity consumption rises sharply. Therefore, considering these factors in the region-based prediction can better simulate the real regional situation in the future.

In this paper, the external module factor is output through two fully connected layers. In feature extraction process, we use one fully connected layer to activate each sub-factor. Then, we use a fully-connected layer to map the features V_{ext} to have the same dimensionality as V_{m3d}. We combine V_{m3d} with the output of the external module. The fused output $\widehat{V}$ is defined in Eq. 8:

$$\widehat{V} = V_{m3d} + V_{ext} \tag{8}$$

$\widehat{V}$ connect the fully connected layer through the Tanh function. Where, Tanh is a hyperbolic tangent function, ensuring that the output value is between -1 and 1. This produces a faster convergence than the standard logistic function in the backpropagation learning process.

4.5. Optimization

In our LMST3D-ResNet, we use the Adam algorithm as the optimizer. The input spatial-temporal correlation information is (defined in 4.1.2) V_c, V_d, and V_s. The feature information is extracted by our model, the value at time t is predicted, and the problem is converted into a regression task. Adam is chosen as the optimizer because the first-order moment estimation and second-order moment estimation of the gradient can be calculated to design independent adaptive learning rates for different parameters. The key to efficient calculations and proper parameter tuning is to take up less memory during training.

4.6. Training model

Our proposed LMST3D-ResNet training process is summarized in Algorithm 1. We analyze the input data and its periodicity, taking a set of historical spatial-temporal correlation features as input. In the specific implementation, we divide different datasets into multiple input signals according to local spatial-temporal features, namely, closeness, period and trend information. Using a 3D convolutional layer and a 3D pooling layer in our framework, all trainable parameters are backpropagated with random initialization. The optimization process uses the Adam function to make maximum use of spatial-temporal correlation information. The proposed model is implemented by TensorFlow and Keras.

Algorithm 1 LMST3D-ResNet Algorithm

Require" Historical observations: $Y^{I,J}_{0,\ldots,n-1}$;
External features: $E^{I,J}_{1,\ldots,n-1}$;
Lengths of *closeness*, *daily* and *trend*: l_c, l_d, l_q;
Daily span: d, Trend span: q.
Ensure: Learned LMST3D-ResNet model.
$D \leftarrow \varnothing$;
//*Local spatial range*;
for $\forall i,j \in I,J$ **do**
for all available time interval $t(1 \leqslant t \leqslant n-1)$ **do**
$V_c \leftarrow= \left[Y^{i,j}_{t-l_c}, Y^{i,j}_{t-(l_c-1)}, \ldots, Y^{i,j}_{t-1}\right]$;
$V_d \leftarrow= \left[Y^{i,j}_{t-l_d\times d}, Y^{i,j}_{t-(l_d-1)\times d}, \ldots, Y^{i,j}_{t-1}\right]$;
$V_s \leftarrow= \left[Y^{i,j}_{t-l_q\times q}, Y^{i,j}_{t-(l_q-1)\times q}, \ldots, Y^{i,j}_{t-1}\right]$;
//Y_t *is the state of the city region at time* t;
(V_c, V_d, V_s, E_t) conveys to D;
end for
end for
//*Training model*;
Initialize all learnable parameters θ in LMST3D-ResNet;
while stopping criteria is not met **do**
Randomly select a batch of instances D_b from D;
Put each V_c, V_d, V_s, E_t of the instance in D_b into the corresponding branch respectively;
Find θ by minimizing the objective with D_b;
end while
return

5. Experiment

In this section, we validate our proposed LMST3D-ResNet on two real datasets. Our proposed LMST3D-ResNet is compared to other existing methods to achieve a comprehensive quantitative assessment.

5.1. Datasets

In the experiment, our proposed LMST3D-ResNet is evaluated on two real datasets (i.e., BJTaxi and MLElectricity). The contents of the dataset are described in detail below.

- **MLElectricity:** The Milan electricity dataset contains 571,392 electricity records for the Milan region from 2013/12/01 to 2013/12/30. In this dataset, Milan is divided into 8×16 regions. The length of each time interval is set to 10 min. Only the inflow record is included, and the outflow record is not included, as described in definition 3. In the experiment, we choose the data from 2013/12/01 to 2013/12/25 as the training data and use the data from 2013/12/26 to 2013/12/30 as the testing data. External module factors include temperatures, weather conditions and holidays.
- **BJTaxi:** The Beijing taxi dataset consists of 4 time periods: from 2013/07/01 to 2013/10/30; from 2014/03/01 to 2014/06/30; from 2015/03/01 to 2015/6/30; from 2015/11/01 to 2016/04/10. In this dataset, Beijing is divided into 32×32 regions. The length of each time interval is set to 30 min. Using definition 2, we obtain the inflow and outflow of the region. In the experiment, we chose from 2013/10/23 to 2013/10/30; from 2014/06/23 to 2014/06/30; from 2015/06/23 to 2015/06/30; and from 2015/04/03 to 2015/04/10; We choose these data as the testing data and other data as the training data. We use the external module factor to be consistent with the MLElectricity dataset.

5.2. Implementation details

5.2.1. Data preprocessing

We use the transformation method to convert the information (i.e., day-of-week, weekend/weekday, holidays and weather conditions) in the external module into binary vectors, which is similar to [6].

We convert the dataset data to the $[0,1]$ scale by min–max normalization, which improves the convergence speed and accuracy of the model. The definition of min–max normalization is as follows:

$$Y_{new} = \frac{Y - Y_{min}}{Y_{max} - Y_{min}}, \tag{9}$$

where Y is the current value of the sample data; Y_{max} is the maximum value of the sample data, and Y_{min} is the minimum value of the sample data; and Y_{new} is the value mapped between $[0,1]$.

We perform min-max normalization on existing methods and compare these methods with our proposed LMST3D-ResNet.

5.2.2. Hyperparameters

We build models based on TensorFlow and Keras. For the MLElectricity dataset, we only consider the electricity inflow, and we set it to the 1 channel. Because there is only one month of data, we made daily predictions for the period. The lengths of *closeness*, *trend* and *period* on MLElectricity are set to 4, 4, and 4, respectively. For the BJTaxi dataset, we consider the inflow and outflow of the vehicle and set it to 2 channels to predict the situation of the regional vehicle. We made daily predictions for the period. The lengths of *closeness*, *trend* and *period* are set to 6, 4, and 4, respectively.

5.2.3. 3D layer parameters

When considering the LMST3D-ResNet structure, we focus on the hyperparameters of the 3D convolutional layer and the 3D pooling layer. We apply the 3D convolutional layer and a residual unit with a batch size of 64, shown in Figure 5. We use the early-stop algorithm to obtain the best verification score on the training model, and the training model sets a fixed number of epochs on the full training data.

For the MLElectricity dataset, its spatial dimension is 8×16, which corresponds to about 8 km $\times$ 16 km rectangles. We set the 3D volume size in all branches to $4 \times 8 \times 16$. The kernel size of the 3D convolutional layer of the first layer of all branches is set to $(3,2,3)$, and the kernel size is set to $(2,2,3)$ in the residual network. The stride of all 3D convolutional layer is set to $(1,1,1)$. The number of 3D convolution filters in all branches is 32. Then, we apply the 3D max-pooling layer followed by the above operation, where the kernel size is $(1,2,2)$. The stride of all 3D max-pooling layer is set to $(1,2,2)$. To reduce the overfitting problem, we use the dropout layer with a dropout rate set to 0.2.

For the BJTaxi dataset, because its spatial dimension is 32×32 larger than the MLElectricity dataset, we set the 3D volume size in all branches to $4 \times 32 \times 32$. In the *closeness* branch, the kernel size of all 3D convolutional layers is $(2,3,3)$. In the *trend* and *period* branches, the 3D convolutional layer kernel size of the first layer is $(2,3,3)$. The kernel size is $(1,3,3)$ in the residual network. The stride of all 3D convolutional layer is set to $(1,1,1)$. The number of 3D convolution filters for the first layer in all branches is 32, and the number of 3D convolution filters in the residual network is 64. The max-pooling and dropout layers are the same as the MLElectricity dataset.

5.2.4. Evaluation metrics

In the experiment, we evaluated our method by the root mean square error (RMSE) and mean average percentage error (MAPE), which is the same as the evaluation of the methods in [34,36,6]. The two evaluation metrics are defined as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{\alpha=1}^{N}\left(\widehat{Y}_t^{i,j} - Y_t^{i,j}\right)^2} \tag{10}$$

$$MAPE = \sqrt{\frac{1}{N}\sum_{\alpha=1}^{N}\frac{|\widehat{Y}_t^{i,j} - Y_t^{i,j}|}{Y_t^{i,j}}} \tag{11}$$

In the city region $i,j \in I,J$ with current time t, $\widehat{Y}_t^{i,j}$ and $Y_t^{i,j}$, respectively, represent the predicted value and the real value, and N is the total number of samples.

5.3. Methods in region-based prediction

We divide the spatial-temporal prediction method into the following three types, and our methods are compared to these methods.

5.3.1. Time-series methods

- **HA:** Historical average(HA) is an estimate of the inflow and outflow of the region based on the average of the historical time interval in the city region.
- **ARIMA [23]:** The autoregressive integrated moving average(ARIMA) is the most common model used in statistical models for time-series prediction. It is widely used because it is relatively simple.

5.3.2. Statistical methods

- **LinUOTD [11]:** LinUOTD uses a simple model structure to avoid the repetitive design of the model and a linear regression model for regularizing spatial-temporal features.
- **XGBoost [29]:** XGBoost is a method of building and extending trees based on approximate tree learning algorithms and sparsity-aware algorithms, which can extend more unknown resources with fewer resources available. This is effective for learning linear spatial-temporal features.

5.3.3. Deep learning methods

- **MultiLayer Perceptron (MLP):** In the method comparison, MLP learns spatial-temporal features for using four fully connected layers of neural networks.
- **ConvLSTM [37]:** ConvLSTM adds a convolutional structure to LSTM. Convolutional layers have the ability to better learn spatial-temporal features.
- **DeepSD [33]:**DeepSD uses artificial neural networks to predict differences in taxi supply and demand patterns. It can be regarded as a kind of MLP-extended model that addresses the supply and demand mode and the external environment information (i.e., weather and traffic information). Note that we have no external environment information; therefore, we have not considered these modules.
- **ST-ResNet [6]:** ST-ResNet is a deep learning framework based on CNNs that join the residual network and modeling traffic vehicle images for different time periods in city regions. From historical images of traffic vehicles, the model captures trends, periods and closeness information of city regions through CNNs. The external module factors are extracted and merged with the output in CNNs. Then, the tan function fuses these related features.
- **STDN [21]:** STDN simulates spatial, temporal, and dynamic correlations by combining local a CNN, an LSTM, and an attention mechanism. In the comparative experiment, we modified the STDN part of the code to fit the dataset. We maintain its original network structure and then compare it. The input of the local CNN is 7×7, the long-term period information is 4, the lengths of the short-term LSTM for the MLElectricity and BJTaxi datasets are set 4 and 6, respectively.
- **MST3D [34]:** MST3D implements multiple 3D CNNs to make full use of low-level spatial-temporal correlation features.

We conduct the Student's t-test. Our proposals of LMST3D and LMST3D-ResNet obtain the lowest RMSE and MAPE on the datasets. We analyze the results of the various methods in Table 1 and Table 2. The traditional time-series prediction methods (i.e., HA and ARIMA) do not achieve good results. They predict future values that depend on historical time records, while they ignore spatial and other external module features.

Spatial prediction methods (i.e., LinUOTD and XGBoost) consider the features of spatial correlation and linear temporal correlation. Therefore, these methods achieve better performance than traditional time-series methods. However, they still fail to capture complex nonlinear temporal correlation and dynamic spatial relationships.

In the deep learning prediction methods (i.e., MLP, DeepSD, ST-ResNet, STDN and MST3D), the following distinctions are noted. MLP learns spatial-temporal features for using fully connected layers of neural networks. However, it does not explicitly simulate the correlation between spatial and temporal features. DeepSD can be regarded as a kind of MLP-extended model that addresses the supply and demand mode and the external environment information. However, it does not consider the temporal and space dependence of the model. ST-ResNet captures the city spatial-temporal correlation features through the CNN powerful feature learning capabilities, but it does not consider local regions. STDN uses local 2D CNNs to extract local spatial features, and it uses an LSTM to extract temporal correlation features. It combines the advantages of the local CNN and LSTM. However, it can only capture high-level layer spatial-temporal correlations without considering low-level layer spatial-temporal correlations. MST3D is proposed implementing multiple 3D CNNs to make full use of low-level spatial-temporal correlation features, but it does not consider local spatial relationships.

In Table 1 and Table 2, our proposed LMST3D-ResNet extracts features for *closeness*, *trend*, and *period* branches in the inflow and outflow of the local region while also considering external module information. LMST3D-ResNet is better than all the baselines. Two different types of datasets achieve the minimum RMSE and MAPE. In the MLElectricity dataset, we obtain an RMSE and MAPE of 6.34 and 20.65%, respectively. In the BJTaxi dataset, the RMSE and MAPE are 15.67 and 14.44%, respectively. Therefore, we can clearly see that our proposed LMST3D and LMST3D-ResNet framework achieve better accuracy, as shown in bold in Table 1.

5.4. Performance evaluation and analysis

Table 1 shows the results of our proposed LMST3D-ResNet compared to the existing methods in RMSE and MAPE. Inflow and outflow methods are considered on the MLElectricity and BJTaxi datasets. Table 2 shows the detailed results of BJTaxi for inflow and outflow.

Table 1
Baseline comparison on MLElectricity and BJTaxi.

Method	MLElectricity		BJTaxi	
	RMSE	MAPE	RMSE	MAPE
HA	14.65	40.37%	57.69	38.34%
ARIMA	10.47	30.28%	22.78	22.13%
LinUOTD	10.05	29.35%	21.23	20.22%
XGBoost	7.56	24.46%	17.84	17.62%
MLP	8.16	25.43%	18.25	17.83%
ConvLSTM	8.31	26.08%	19.54	18.63%
DeepSD	7.43	23.86%	18.07	17.71%
ST-ResNet	7.18	22.61%	16.89	15.48%
STDN	6.93	21.89%	16.65	15.27%
MST3D	6.51	21.15%	15.99	14.78%
LMST3D	**6.47**$^{\Delta}$	**20.87**%$^{\Delta}$	**15.84**$^{\Delta}$	**14.73**%$^{\Delta}$
LMST3D-ResNet	**6.34**$^{\Delta\Delta}$	**20.65**%$^{\Delta\Delta}$	**15.67**$^{\Delta\Delta}$	**14.44**%$^{\Delta\Delta}$

$\Delta\Delta(\Delta)$ means the result is significant according to Students T-test at level 0.01 (0.05) compared to MST3D.

Table 2
Inflow and outflow results on BJTaxi.

Methods	Inflow		Outflow	
	RMSE	MAPE	RMSE	MAPE
HA	57.57	37.76%	57.89	39.68%
ARIMA	22.58	22.12%	22.96	22.19%
LinUOTD	21.19	20.02%	21.44	20.33%
XGBoost	17.61	17.42%	18.23	17.69%
MLP	18.23	17.54%	18.30	18.21%
ConvLSTM	19.29	18.55%	19.98	18.72%
DeepSD	17.95	17.29%	18.31	17.94%
ST-ResNet	16.74	15.01%	17.01	15.78%
STDN	16.43	15.12%	16.78	15.44%
MST3D	15.98	14.71%	16.11	14.85%
LMST3D	**15.78**	**14.67%**	**15.87**	**14.78%**
LMST3D-ResNet	**15.64**	**14.36%**	**15.71**	**14.48%**

5.5. Performance of multiple different factors

We evaluated the performance of our method through branches of different factors on the MLElectricity and BJTaxi datasets.

5.5.1. Our proposed LMST3D performance

Figure 6 shows the RMSE and MAPE in the MLElectricity dataset. Figure 7 shows the RMSE and MAPE in the BJTaxi dataset. The processing method is as follows:

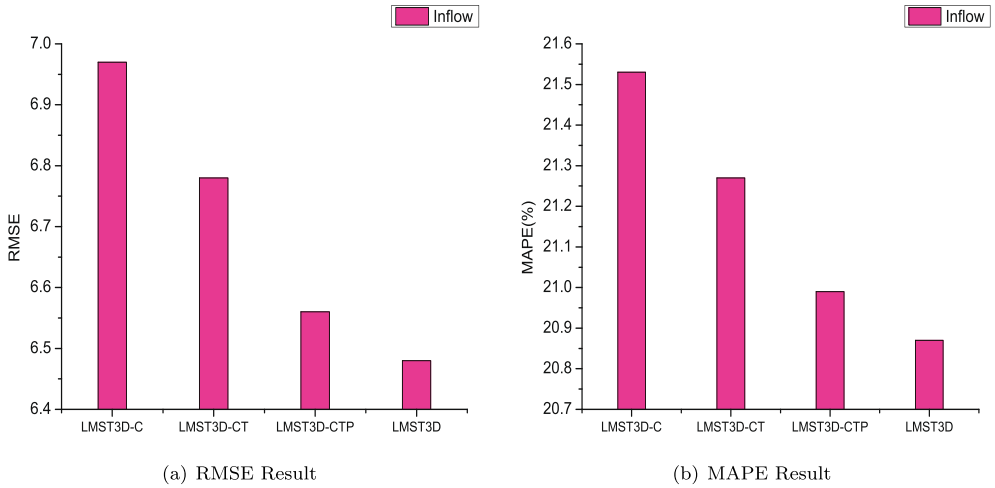
- **LMST3D-C:** In this method, we only consider the spatial-temporal features of the *closeness* branch.
- **LMST3D-CT:** In this method, our framwork uses *closeness* and *trend* branches only.
- **LMST3D-CTP:** In this method, our framwork uses *closeness*, *trend* and *period* branches.
- **LMST3D:** Our framework considers *closeness*, *trend*, *period* and *external* branches.

According to Figure 6 and 7, LMST3D-C learns local spatial-temporal features of the *closeness* branch. LMST3D-CT and LMST3D-CTP further improve performance by adding *trend* and *period* branches. LMST3D adds an *external* branch that combines the different factors. It further decreases the RMSE and MAPE values compared to other benchmarks. LMST3D proves that considering multiple factors can help improve the accuracy of city region predictions.

5.5.2. Our proposed LMST3D-ResNet performance

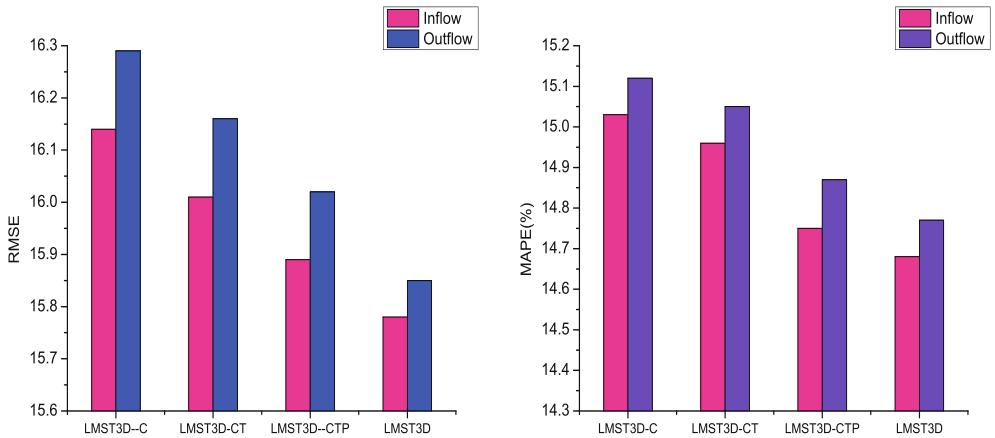
Figure 8 shows the RMSE and MAPE in the MLElectricity dataset. Figure 9 shows the RMSE and MAPE in the BJTaxi dataset. We added a residual network to the LMST3D, and the other processing is similar to the LMST3D as follow:

- **LMST3D-ResNet-C:** In this method, we only consider the spatial-temporal features of the *closeness* branch.
- **LMST3D-ResNet-CT:** In this method, our framework uses *closeness* and *trend* branches only.



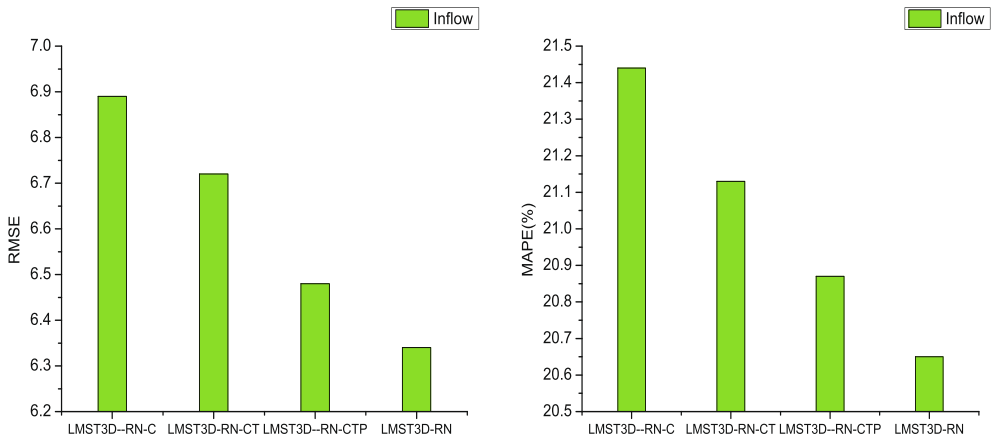
(a) RMSE Result (b) MAPE Result

Fig. 6. Different factor results for LMST3D on MLElectricity.



(a) RMSE Result (b) MAPE Result

Fig. 7. Different factor results for LMST3D on BJTaxi.



(a) RMSE Result (b) MAPE Result

Fig. 8. Different factor results for LMST3D-ResNet on MLElectricity.

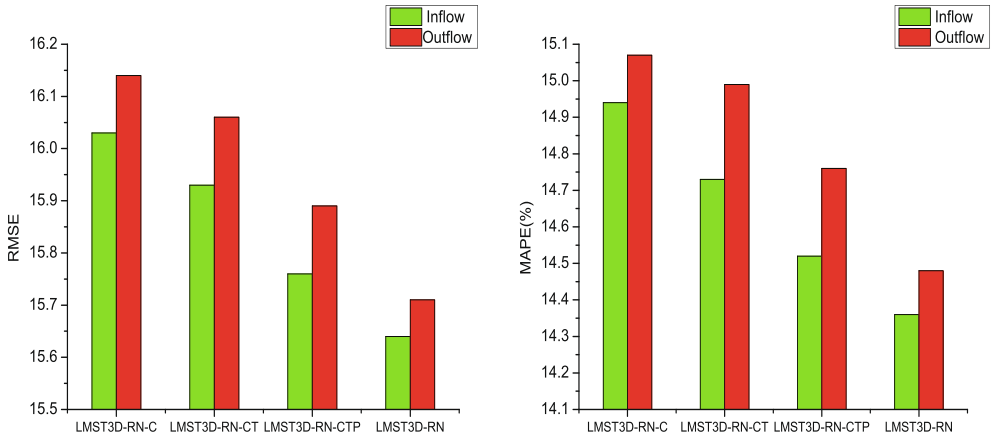


Fig. 9. Different factor results for LMST3D-ResNet on BJTaxi.

Table 3
Running time of different methods.

Methods	MLElectricity		BJTaxi	
	Training time (s)	Testing time (s)	Training time (s)	Testing time (s)
STDN	18980	89.8	379600	207.4
LMST3D-ResNet	8680	37.6	220906	97.6
LMST3D	7426	34.3	196384	70.5

- **LMST3D-ResNet-CTP:** In this method, our framework uses *closeness*, *trend* and *period* branches.
- **LMST3D-ResNet:** Our framework considers *closeness*, *trend*, *period* and *external* branches.

In Figures 8 and 9, we added the residual network. We can see that the performance is further improved under different branch conditions. As we described in 4.2, we obtain the lowest RMSE and MAPE compared to other benchmarks.

5.6. Time complexity on different methods of local regions

In Table 3, we compare the runtime of training and testing on existing methods of local region spatial-temporal features. To compare performance fairly, we evaluate these methods on a single P100 GPU [38]. We see that STDN has the longest running time in training and testing, and STDN uses a local 2D CNNs + LSTM to predict each region. STDN at high-level layers combines local spatial features with temporal features, thus depleting more runtime. However, the method adopting 3D CNNs is implemented to extract features directly, so the running time is relatively small.

In our comparison of LMST3D and LMST3D-ResNet, we can see that LMST3D runs utlizing less time than LMST3D-ResNet. This is mainly due to the LMST3D-ResNet cross-layer connection operation, so the BJTaxi dataset of 32×32 is more obvious than the MLElectricity dataset of 8×16.

In summary, our proposed LMST3D and LMST3D-ResNet significantly improved runtime benchmarks. LMST3D-ResNet consumes some runtime compared to LMST3D, but it is still in an acceptable range and LMST3D-ResNet has relatively better performance.

6. Conclusions

Region-based predictions are critical to building smart cities, which is challenging for a variety of factors (i.e., climate, events and weekends). This paper proposes to learn the spatial-temporal features of the region-based prediction, taking into account the correlations of local region spatial and temporal features and the impact of external module factors. In our proposed model, a novel region-based information extraction mechanism and an end-to-end multiple spatial-temporal dependency learning structure are designed for local regions. We propose the LMST3D and LMST3D-ResNet frameworks for region-based prediction and conduct experiments on two different types of real datasets. Experimental results show that the proposed framework is superior to the most advanced baseline. In future work, we will further explore a better fusion

framework to learn spatial-temporal features. More semantic information (i.e., graph-structured information) will be added to the framework through graph learning.

CRediT authorship contribution statement

Yibi Chen: Conceptualization, Methodology, Validation, Writing - original draft. **Xiaofeng Zou:** Methodology, Validation, Writing - review & editing. **Kenli Li:** Supervision, Validation, Writing - review & editing. **Keqin Li:** Supervision, Software. **Xulei Yang:** Validation, Writing - review & editing. **Cen Chen:** Formal analysis, 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.

Acknowledgments

The research was partially funded by the National Key R&D Program of China (Grant No. 2018YFB1003401), the National Outstanding Youth Science Program of the National Natural Science Foundation of China (Grant No. 61625202), the National Natural Science Foundation of China (Grant No. 61902120), the Postdoctoral Science Foundation of China (Grant No. 2019M662768, 2019TQ0086) and the International (Regional) Cooperation and Exchange Program of the National Natural Science Foundation of China (Grant No. 61661146006, 61860206011).

References

[1] B. Mocanu, F. Pop, A. Mihaita, C. Dobre, A. Castiglione, Data fusion technique in spider peer-to-peer networks in smart cities for security enhancements, Inf. Sci. 479 (2019) 607–621.
[2] D. Li, L. Deng, B.B. Gupta, H. Wang, C. Choi, A novel cnn based security guaranteed image watermarking generation scenario for smart city applications, Inf. Sci. 479 (2019) 432–447.
[3] C. Chen, K. Li, A. Ouyang, Z. Zeng, K. Li, Gflink: An in-memory computing architecture on heterogeneous cpu-gpu clusters for big data, IEEE Trans. Parallel Distrib. Syst. 29 (6) (2018) 1275–1288.
[4] C. Chen, K. Li, A. Ouyang, Z. Tang, K. Li, Gpu-accelerated parallel hierarchical extreme learning machine on flink for big data, IEEE Trans. Syst., Man, Cybern.: Syst. 47 (10) (2017) 2740–2753.
[5] O. Castillo, F. Kutlu, Ö. Atan, Intuitionistic fuzzy control of twin rotor multiple input multiple output systems, J. Intell. Fuzzy Syst. 38 (1) (2020) 821–833.
[6] J. Zhang, Y. Zheng, D. Qi, Deep spatio-temporal residual networks for citywide crowd flows prediction, in: Thirty-First AAAI Conference on Artificial Intelligence, 2017.
[7] X. Li, G. Pan, Z. Wu, G. Qi, S. Li, D. Zhang, W. Zhang, Z. Wang, Prediction of urban human mobility using large-scale taxi traces and its applications, Front. Comput. Sci. 6 (1) (2012) 111–121.
[8] L. Moreira-Matias, J. Gama, M. Ferreira, J. Mendes-Moreira, L. Damas, Predicting taxi-passenger demand using streaming data, IEEE Trans. Intell. Transp. Syst. 14 (3) (2013) 1393–1402.
[9] S. Shekhar, B.M. Williams, Adaptive seasonal time series models for forecasting short-term traffic flow, Transp. Res. Rec. 2024 (1) (2007) 116–125.
[10] D. Deng, C. Shahabi, U. Demiryurek, L. Zhu, R. Yu, Y. Liu, Latent space model for road networks to predict time-varying traffic, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2016, pp. 1525–1534.
[11] Y. Tong, Y. Chen, Z. Zhou, L. Chen, J. Wang, Q. Yang, J. Ye, W. Lv, The simpler the better: a unified approach to predicting original taxi demands based on large-scale online platforms, in: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, ACM, 2017, pp. 1653–1662.
[12] B. Pan, U. Demiryurek, C. Shahabi, Utilizing real-world transportation data for accurate traffic prediction, in: 2012 IEEE 12th International Conference on Data Mining, IEEE, 2012, pp. 595–604.
[13] F. Wu, H. Wang, Z. Li, Interpreting traffic dynamics using ubiquitous urban data, in: in: Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, ACM, 2016, p. 69.
[14] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436.
[15] S. Zhou, X. Liu, Q. Liu, S. Wang, C. Zhu, J. Yin, Random fourier extreme learning machine with l2, 1-norm regularization, Neurocomputing 174 (2016) 143–153.
[16] S. Zhou, X. Liu, M. Li, E. Zhu, L. Liu, C. Zhang, J. Yin, Multiple kernel clustering with neighbor-kernel subspace segmentation, IEEE Trans. Neural Networks Learn. Syst..
[17] J. Zhang, Y. Zheng, D. Qi, R. Li, X. Yi, Dnn-based prediction model for spatio-temporal data, in: Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, ACM, 2016, p. 92.
[18] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Adv. Neural Inform. Process. Syst. (2012) 1097–1105.
[19] R. Yu, Y. Li, C. Shahabi, U. Demiryurek, Y. Liu, Deep learning: a generic approach for extreme condition traffic forecasting, in: Proceedings of the 2017 SIAM International Conference on Data Mining, SIAM, 2017, pp. 777–785.
[20] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[21] H. Yao, X. Tang, H. Wei, G. Zheng, Y. Yu, Z. Li, Modeling spatial-temporal dynamics for traffic prediction, arXiv preprint arXiv:1803.01254..
[22] A. Buzachis, A. Celesti, A. Galletta, M. Fazio, G. Fortino, M. Villari, A multi-agent autonomous intersection management (ma-aim) system for smart cities leveraging edge-of-things and blockchain, Inform. Sci..
[23] B.M. Williams, L.A. Hoel, Modeling and forecasting vehicular traffic flow as a seasonal arima process: theoretical basis and empirical results, J. Transp. Eng. 129 (6) (2003) 664–672.
[24] G.E. Box, G.M. Jenkins, G.C. Reinsel, G.M. Ljung, Time series analysis: forecasting and control, John Wiley & Sons, 2015.
[25] H. Lütkepohl, New introduction to multiple time series analysis, Springer Science & Business Media, 2005.
[26] X. Tang, H. Yao, Y. Sun, C. Aggarwal, P. Mitra, S. Wang, Joint modeling of local and global temporal dynamics for multivariate time series forecasting with missing values, arXiv preprint arXiv:1911.10273..

[27] J. Soto, O. Castillo, P. Melin, W. Pedrycz, A new approach to multiple time series prediction using mimo fuzzy aggregation models with modular neural networks, Int. J. Fuzzy Syst. 21 (5) (2019) 1629–1648.
[28] J. Soto, P. Melin, O. Castillo, A new approach for time series prediction using ensembles of it2fnn models with optimization of fuzzy integrators, Int. J. Fuzzy Syst. 20 (3) (2018) 701–728.
[29] T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, ACM, 2016, pp. 785–794.
[30] T. Idé, M. Sugiyama, Trajectory regression on road networks, in: Twenty-Fifth AAAI Conference on Artificial Intelligence, 2011.
[31] J. Zheng, L.M. Ni, Time-dependent trajectory regression on road networks via multi-task learning, in: Twenty-Seventh AAAI Conference on Artificial Intelligence, 2013.
[32] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, Y. Wang, Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction, Sensors 17 (4) (2017) 818.
[33] D. Wang, W. Cao, J. Li, J. Ye, Deepsd: Supply-demand prediction for online car-hailing services using deep neural networks, in: 2017 IEEE 33rd international conference on data engineering (ICDE), IEEE, 2017, pp. 243–254..
[34] C. Chen, K. Li, S.G. Teo, G. Chen, X. Zou, X. Yang, R.C. Vijay, J. Feng, Z. Zeng, Exploiting spatio-temporal correlations with multiple 3d convolutional neural networks for citywide vehicle flow prediction, in: 2018 IEEE International Conference on Data Mining (ICDM), IEEE, 2018, pp. 893–898.
[35] Jan W-P, Liu X. Deep residual learning for image recognition; 2015..
[36] D. Deng, C. Shahabi, U. Demiryurek, L. Zhu, Situation aware multi-task learning for traffic prediction, ICDM (2017) 81–90.
[37] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, W.-C. Woo, Convolutional lstm network: A machine learning approach for precipitation nowcasting, Adv. Neural Inform. Processing Syst. (2015) 802–810.
[38] C. Chen, K. Li, A. Ouyang, K. Li, Flinkcl: an opencl-based in-memory computing architecture on heterogeneous cpu-gpu clusters for big data, IEEE Trans. Comput. 67 (12) (2018) 1765–1779.