Exploiting Dynamic Spatio-Temporal Patterns for Predicting Citywide Traffic Crowd Flows Using An Attention Based Deep Hybrid Neural Networks

Ahmad Ali[a,1,], Yanmin Zhu[b,1], Qiuxia Chen[c,1], Lanqing Yang[d,1,1], Jiadi Yu[e,1,1]

[a] *Shanghai Jiao Tong University, China*
[b] *Shanghai Jiao Tong University, China*
[c] *Shenzhen Polytechnic, China*
[d] *Shanghai Jiao Tong University, China*
[e] *Shanghai Jiao Tong University, China*

Abstract

Predicting the citywide accurate traffic crowd flows is a critical issue for practitioners and researchers in the field of intelligent transportation systems (ITS). However, its key challenge lies in how to model multiple complicated spatial dependencies (distant and nearby) and temporal laws with an external factors (e.g., events, weather and holidays). To address this problem, we propose a unified deep hybrid spatio-temporal dynamic neural network namely DHSTNet, to predict both inflows and outflows in each and every region of a city. In particular, our DHSTNet model decomposed into four components, i.e.,, closeness influence taking the instantaneous changes of traffic crowd flows, period influence regularly identifying daily variations of traffic flows, weekly influence identifying the weekly patterens of traffic flows and external component acquiring the external factors. Further, we design a branch of deep hybrid recurrent convolutional neural network units to depict the first three temporal properties, i.e.,, closeness, period influence and weekly influence. The external components are feed into two fully connected neural networks. Our proposed model assign different weights to different branches and then integrate the output of four com-

[⋆] This paper is an extended version of [8], which has been already accepted for presentation at 2019 IEEE 25th international Conference on Parallel and Distributed Systems (ICPADS).

[*] Corresponding author

Email addresses: ahmadali@sjtu.edu.cn (Ahmad Ali), yzhu@sjtu.edu.cn (Yanmin Zhu)

Preprint submitted to Journal of LaTeX Templates *April 7, 2020*

ponents to generate the final prediction results. Extensive experiments based on two large-scale real-world datasets well demonstrate the superiority of our model over the existing state-of-the-art baselines. Moreover, to verify the generalization of our our model, we also apply the attention-based mechanism with our previous model called as Att-DHSTNet to predict citywide short-term traffic crowd flows and show its notable performance in this traffic flows prediction task.

Keywords: Spatio-temporal dynamics, Deep learning, Long short term memory, crowd flows prediction, Convolutional neural network, Attention-based mechanism.

1. Introduction

City is the cornerstone of advance people living and constantly move from rural regions to urban regions with urbanization. Traffic crowd flow prediction is a spatio-temporal problem in urban computing environments. It is crucial to traffic management, risk assessment, public safety and environmental pollution for citywide planners [1]. Traffic crowd flow is one of the important activities in urban traffic prediction. As more traffic crowd data are collected from traffic cameras [2], mobile devices [3], and GPS devices [4], the problem is being more voluminous and complex. However, urban traffic crowd flow prediction on traffic data is a challenging task. Therefore, an accurate and scalable traffic prediction models are desirable for handling the computational complexity of large traffic data. On 31st December 2014, a deadly stampede occurred in the close area of Chen Yi Square Bund in Shanghai, China, where a crowd of almost 300,000 people had gathered for celebration of the new year. About 49 people were injured, 36 were killed, and the number of seriously injured people was 13. If one can predict accurately in advance the coming flow of crowds and compare it to the crowd capacity of the region, such tragedies could be mitigated.

In this paper, we aim to predict two types of future urban crowd flows, i.e.,, inflow and outflow. The inflow and outflow of traffic crowds is the total number people entering and leaving the region, respectively, as shown in figure 1. Simultaneously predicting the inflow and outflow in each area is a crucial task. Inflows and outflows in each area can be basically affected by three important complex factors including:

(1) Spatial dependencies: Figure 1 shows that the inflow of region r_2 is affected by the outflow of the nearby areas i.e., r_1 and r_3. Likewise, the outflow of region r_2 also affects the inflow of its nearby regions, while the inflow of r_2 might affect its outflow as well.

(2) Temporal dependencies: Traffic flows change gradually in continuous time and also show periodic patterns. for instance, the traffic congestion occurring at 8am will definitely affect that of 9am. The morning and afternoon peak hours may appear similar patterns on weekdays.

(3) External factors: The urban traffic flows can be directly affected by external factors, including weather situations, season of the year, road works and other events. For instance, a thunderstorm can also affects the traffic flows speed on roads as well as changes the traffic flows of regions.

In recent years, a notable accomplishment have been achieved for urban traffic crowd flow prediction based on deep learning with high dimensional spatio-temporal data [3][5][6][7]. Based on these works, a city is divided into a grid map based on the longitude and latitude respectively. In [3] a CNN method called ST-ResNet is proposed, which is better than previous neural network models.

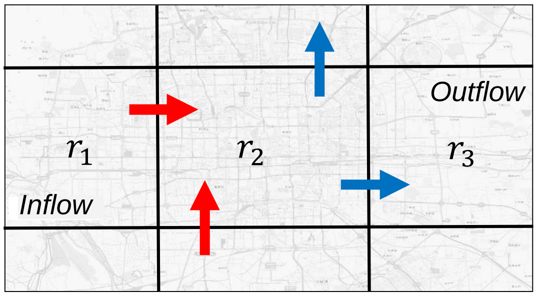


Figure 1: Inflow and outflow of a particular region.

For spatial dependency, the ST-ResNet model uses a residual neural network architecture to extract influences from distant areas. However, such information is not much important for short term prediction. On the other hand, the ST-ResNet model requires a manual classification of temporal dependency, which makes it difficult to define the temporal dependency. This approach is unable to learn the different dependency strength of different time intervals. Therefore, temporal dependencies outweigh the influences coming from areas that are far away from the targeted areas. However, these methods still fail to model simultaneously the spatial and temporal features and dynamic correlations of traffic flows data. Different models have been proposed for traffic prediction. Deep learning models show better prediction accuracy as compared to other traditional approaches.

In order to solve the challenges mentioned above, we propose a novel deep learning spatio-temporal model to simultaneously predict traffic crowd flow at each location on the traffic network. It is a deep hybrid spatio-temporal dynamic neural networks, namely DHSTNet. this DHSTNet model collectively predict both inflow and outflow of the regions. It is based on spatial and temporal network, which handles spatial and temporal correlation via CNN and LSTM, respectively. Based on the proposed DHSTNet, we further developed an attention-based mechanism called as (Att-DHSTNet), which adaptively and simultaneously captures all dependencies and thus are more useful as the spatio-temporal dependencies can interrelate to each other. To the best of our knowledge, the proposed Att-DHSTNet method is the first work that can synchronously measure both dependencies. The principal contributions of this paper are summarized as follows:

- We propose a novel deep hybrid spatio-temporal dynamic neural networks, which dynamically learn the spatio-temporal correlations of urban traffic crowd flows data.
- We design a framework, called as DHSTNet, considering spatio-temporal dependencies with an external factors. We also classify the temporal prop-

erties of urban crowd flows into three main classes, including (closeness, period influence, and weekly influence), respectively. DHSTNet combines the output of three components with the external components and simultaneously assigns different weights to different branches.

- Extensive evaluations on two real-world datasets including taxi data of Beijing and bike data of New York City. The experimental results demonstrate that our proposed model achieves the better prediction performances than the existing state-of-the-art baselines.

- We further developed an attention mechanism with DHSTNet, called as Att-DHSTNet, which simultaneously handle sophisticated and dynamic spatio-temporal dependencies of traffic flows data. A spatial attention is used to depict the spatial complex correlation between different regions, while the temporal attention is applied to extract the dynamic temporal patterens between different time steps.

The earlier version of this paper published in [8]. In this current work, we design an attention-based mechanism with our previous DHSTNet model called as Att-DHSTNet in Section 3, to collectively predict spatio-temporal correlations of citywide crowd flows prediction. Further, we conduct more comprehensive experiments and compare with more state-of-the-art baselines under different experimental settings with attention and without attention data, impact of different configurations including (impact of depth network, effect of filter numbers and kernel sizes, and with external and without external data) in Section 4. We also explored the related work in Section 5.

The rest of our paper is structured as follows. The problem is formulated in Section 2. The detailed design of the DHSTNet model with attention mechanism is presented in Section 3. Extensive experiments and comparisons are discussed in Section 4. We review the related work of urban traffic crowd flow in Section 5. Finally, we conclude this paper in Section 6.

2. Preliminaries and Problem Formulation

2.1. Citywide Traffic Crowd flows Prediction

In this section, first we describe some basic notations of urban crowd flow and then define the citywide traffic crowd flows prediction problem.

Definition 1 (Citywide Region Partition): Our target is to predict both inflow and outflow of crowds within a give area. As following the existing research works [9][10], we divide the entire city into (I x J) grid-based map, which is logically divided by latitude and longitude.

Definition 2 (Traffic inflow/outflow): In [11], let S be a set of trajectories at time interval t. For region (i , j) means the i row and the j column. The inflow and outflow of the traffic flows at time t are defined as follows.

$$x_t^{in,i,j} = \sum_{T_r \in S} |\{t > 1 | g_{t-1} \notin (i,j) \wedge g_t \in (i,j)\}| \tag{1}$$

$$x_t^{out,i,j} = \sum_{T_r \in S} |\{t \geq 1 | g_t \in (i,j) \wedge g_{t+1} \notin (i,j)\}| \tag{2}$$

where T_r:$g_1 \rightarrow g_2 \rightarrow$,..., $\rightarrow g_{|Tr|}$ is a trajectory in S, while g_t denotes the geospatial coordinate and $g_t \in$ (i, j) indicates that the point of g_t lies within the grid map (i, j).

Formally, for citywide urban traffic crowd flows the multi-channel image is defined in definition 3.

Definition 3: At given time interval t, regional inflow and outflow in all regions of ($I \times J$) can be represented as a 3D tensor $\mathbf{X_t} \in \mathbb{R}^{I \times J \times 2}$, where $(\mathbf{X_t})_0$ denotes the inflow matrix while $(\mathbf{X_t})_1$ represents the outflow matrix respectively.

Problem 1 (Traffic Crowd flows Prediction): Given the observations of urban crowd flows, denoted by T = 1,2,...,t-1. We aim to predict both inflow and outflow of every region of the entire city map at the next time interval t.

3. Attention-Based Deep Hybrid Spatio-Temporal Dynamic Neural Networks

In this section, we describe the details of our previous proposed DHSTNet model. Furthermore, we design an attention based mechanism with DHSTNet in this paper called as Att-DHSTNet. It consists of three steps, i.e., modeling, training, and testing.

Figure 2 illustrates the overall framework of our proposed Att-DHSTNet model, which is comprised of four main components, i.e., closeness, period influence, weekly influence, and external factors of historical data, respectively. As mentioned in Section 2, a city is divided into a grid-based map. Therefore, the traffic crowd flows can be changed into inflow and outflow as a 2-channel image-like matrix at each time interval by using the approach discussed in definition (1) and (2), respectively. We then divide the time stream into different time intervals, which makes the traffic crowd flow sequence data like the video stream, then we feed the crowd flows sequence data into the first three components separately to describe the above temporal correlation: closeness, period influence and, weekly influence, respectively.

The first three components share the same neural network structure with a long short term memory followed by a convolutional neural network and each of them consists a fully connected layer and several spatio-temporal blocks. In every spatio-temporal block, there are ConvLSTM module and attention module. To optimize the efficiency of the training process, we employ a residual based framework to each component. for each component, the spatial dependencies of the nearby regions are captured by CNN. To train the model, we consider each area as an independent sample. In our model, we study the spatial dependencies of adjacent areas. We use the LSTM model to capture the temporal dependencies of every sequence of temporal data. To achieve the better temporal correlations of urban traffic crowd flows at different time intervals, we applied an attention based ConvLSTM model to handle the global feature series. The we reshaped the generated spatio-temporal feature and feed into an softmax activation function to obtain the prediction. In the external branch, we extract manually some features from the external datasets, which include vacation, weather condition, wind, etc. After that, the extracted features are feed into a two layers fully connected (FC) neural network. Based on the parameter matrices, the output of these three components has fused as $\mathbf{X_{fusion}}$, which assigns different weights to the outputs of different properties in different regions. Furthermore, $\mathbf{X_{fusion}}$ is combined with the result of an external branch as $\mathbf{X_{ext}}$. As a result, the spatio-temporal features and external features are fused them together. Finally, we apply the FC neural network to calculates the cross-entropy loss in traffic crowd flows prediction.

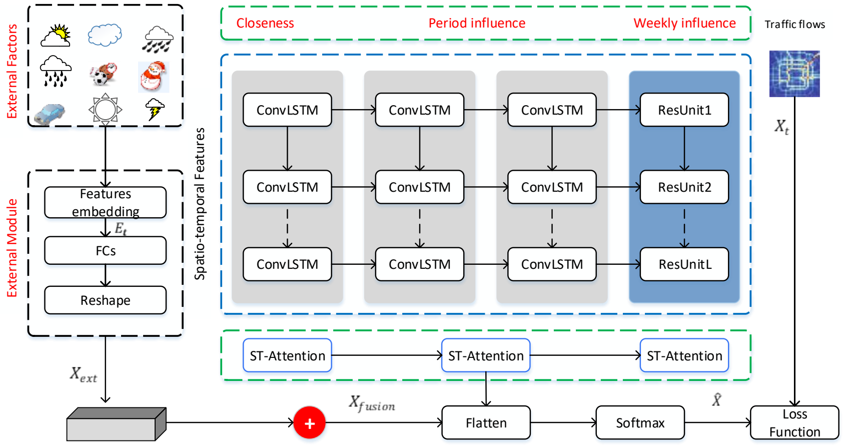


Figure 2: The architecture of Att-DHSTNet, Fully connected (FC), Spatio-temporal attention (ST-Attention), Convolution and Long Short Term Memory (ConvLSTM)

3.1. Spatial And Multiple Temporal Attention Mechanism

In our model, we proposed a novel spatio-temporal attention-based mechanism to collectively capture the dynamic spatio-temporal correlations on the urban traffic crowd flows network as shown in figure 2. It mainly consists of two types of attentions i.e., spatial and multiple temporal attention.

3.2. Modeling Spatio-temporal Correlation Dependencies with Attention Mechanism

In the spatial domain, the urban traffic flow conditions of diverse areas have impact among each other and the common impact is exceedingly dynamic. Here, we apply an attention mechanism [12] to utilize the dynamic correlations among nodes in spatial domain. A given city is divided into $(I \times J)$ regions at given time interval t, the urban traffic flows conditions can be denoted as a tensor $\mathbf{X_t}$ $\in \mathbb{R}^{i \times j \times k}$, where k represents the amount of traffic variables. The generated tensor can be considered as a notable multi-channel image , including height with pixels i, width with pixels j and each pixel of channels k. This kind of image captures spatial correlations of urban traffic crowd flows. Particularly, the multi-channel image can jointly capture the spatial and temporal correlations of traffic flows by keeping the value of k=2.

3.3. Modeling Multiple Temporal Dependencies with Attention Mechanism

It is obvious from the above cases that, the temporal dependencies and their correlation have important impacts on urban traffic conditions, i.e.,, closeness, period influence, and weekly influence respectively, but degrees of influence may not be the same. Moreover, we utalize an attention-based mechanism to adaptively connect different significance to data. We take the temporal dependency of three properties including closeness, period influence, and weekly influence respectively. The closeness component adopts a few two-channel images of intervals in recent times that are used to depict closeness temporal dependency. On top of L^{th} residual unit, we append a convolutional LSTM layer with convolution and L residual units as shown in figure 2. The output of closeness component is $\mathbf{X}_{c,t}^{(L+2)}$. At any time t, let a recent fragment be $[X_{t-l_c}, X_t - (l_{c-1}), ..., X_{t-1}]$ (note that each element here as a 3D tensor $\in \mathbb{R}^{2 \times I \times J}$). first we integrate them with the first axis (i.e.,, time interval) as one tensor $\mathbf{X}_{c,t}^{(0)} \in \mathbb{R}^{l_c \times 2 \times I \times J}$, which is followed by a Convolutional LSTM layer as follows:

$$\begin{aligned}
i_t =&\sigma\left(W_{xi} * \mathbf{X}_{c,t}^{(0)} + W_{hi} * \mathbf{X}_{c,t-1}^{(1)} + W_{ci} \circ \mathcal{C}_{t-1} + b_i\right) \\
f_t =&\sigma\left(W_{xf} * \mathbf{X}_{c,t}^{(0)} + W_{hf} * \mathbf{X}_{c,t-1}^{(1)} + W_{cf} \circ \mathcal{C}_{t-1} + b_f\right) \\
\mathcal{C}_t =&f_t \circ \mathcal{C}_{t-1} + i_t \circ \tanh\left(W_{xc} * \mathbf{X}_{c,t}^{(0)} + W_{hc} * \mathbf{X}_{c,t-1}^{(1)} + b_c\right) \\
o_t =&\sigma\left(W_{xo} * \mathbf{X}_{c,t}^{(0)} + W_{ho} * \mathbf{X}_{c,t-1}^{(1)} + W_{co} \circ \mathcal{C}_t + b_o\right) \\
\mathbf{X}_{c,t}^{(1)} =&o_t \circ \tanh\left(\mathcal{C}_t\right)
\end{aligned} \tag{3}$$

Where $*$ denotes the convolutional operation and $\circ$ represents the Hadamard product, W_{xi}, W_{xf}, W_{cf}, W_{hf}, W_{xc}, W_{hc}, b_i, b_f, b_c, b_o are denotes the learnable parameters in the first layer.

Similarly, we can formulate the period influence and weekly influence components using the above operations, respectively. Assume that l_d is a time interval from the period fragment and the period is d. Therefore, the dependent sequence of daily period influence is $[X_{t-l_d\times d}, X_{t-(l_{d-1})\times d}, ..., X_{t-1}]$. The output of the period influence is $\mathbf{X}_{d,t}^{(L+2)}$. The weekly influence dependent sequence is $[X_{t-l_r\times r}, X_{t-(l_{r-1})\times r}, ..., X_{t-1}]$ and the output of the weekly influence is $\mathbf{X}_{r,t}^{(L+2)}$, where l_r denotes length of weekly influence dependent sequence and r is the weekly span. In our comprehensive implementation, d=1 that shows the day-level periodicity, while r=1 that shows the weekly-level movement of the network.

After we apply attention mechanism, the outputs of three dependent components are as follows:

$$\mathbf{X}_{c,t}^{'} = \lambda_3 \circ (\lambda_1 \circ X_{c,t}^{L} + \lambda_2 \circ X_{c,t}^{L+1}) + \lambda_4 \circ X_{c,t}^{L+2} \tag{4}$$

$$\mathbf{X}_{d,t}^{'} = \lambda_3 \circ (\lambda_1 \circ X_{c,t}^{L} + \lambda_2 \circ X_{c,t}^{L+1}) + \lambda_4 \circ X_{c,t}^{L+2} \tag{5}$$

$$\mathbf{X}_{r,t}^{'} = \lambda_3 \circ (\lambda_1 \circ X_{c,t}^{L} + \lambda_2 \circ X_{c,t}^{L+1}) + \lambda_4 \circ X_{c,t}^{L+2} \tag{6}$$

Where $\mathbf{X}_{c,t}^{'}$, $\mathbf{X}_{d,t}^{'}$, $\mathbf{X}_{r,t}^{'}$ denotes the outputs closeness, period influence, and weekly influence respectively. As we mentioned above, all areas influenced by multiple temporal dependencies and degrees of influence are not completely the same. Motivated by the observations, we consider a novel parametric-tensor-based mechanism to fuse the first three components (closeness, period influence and weekly influence) of figure 2 as follows.

$$\mathbf{X_{fusion}} = \mathbf{W_c} \circ \mathbf{X}_{c,t}^{'(L+2)} + \mathbf{W_d} \circ \mathbf{X}_{d,t}^{'(L+2)} + \mathbf{W_r} \circ \mathbf{X}_{r,t}^{'(L+2)} \tag{7}$$

Where $\mathbf{X_{fusion}}$ represents the fused features, $\circ$ represent the Hadamard product (multiplication element-wise for tensors), while $\mathbf{W_c}$, $\mathbf{W_d}$, and $\mathbf{W_r}$ denotes the learnable parameters that arrange the degrees influenced by different branches.

3.4. External Component Structure

The urban traffic crowd flows can be affected by several external complex factors such as some special events, weather conditions, etc. For example, vacation like Christmas and Chinese New Year celebration can have heavy traffic crowd flows compared to non-vacation. In this work, our main concentration is on special events, vacation events and meta-data including (weekday, day-of-the-week and weekend) as well as weather conditions. To predict the urban traffic crowd flows in time interval t, we can calculate the holiday event and meta-data, but hard to determine the weather at future time interval t. Instead, to solve this issue, we can use the weather forecasting condition at the time interval t or approximately weather situation from historical weather data at a time interval (t-1). Formally, we stack two layers fully connected (FC) layers at E_t, for each sub-factor the first one can be seen as an embedding layer followed by an activation function, while the another FC layer is used to map from low to high dimensions that have the same shape as $\mathbf{X_t}$.

Fusing external component: In this step, we directly merge the first three components output with that of the external properties, as shown in figure 2. The fused output $\hat{\mathbf{X_t}}$ of the three components and external properties is defined in Eq. 8. We use hyperbolic tangent function to make sure that the result values are between [-1,1].

$$\mathbf{\hat{X}_t} = (\mathbf{X_{fusion}} + \mathbf{X_{ext}}) \tag{8}$$

Our Att-DHSTNet model can be trained to predict $\mathbf{X_t}$ from four properties, i.e.,, closeness, period influence, weekly influence and external factor respectively by reducing the Mean Squared Error (MSE) value between the ground flow and predicted flow matrix in the t^{th} time interval.

$$Loss(\theta) = \|\mathbf{X_t} - \mathbf{\hat{X}_t}\|_2^2 \tag{9}$$

Where θ includes $\mathbf{W_c}$, $\mathbf{W_d}$ and $\mathbf{W_r}$ and other learnable parameters in the Att-DHSTNet model.

3.5. Model Training

The training process of the Att-DHSTNet model is outlined in algorithm 1. All the trainable parameters in the proposed Att-DHSTNet model are initialized randomly and optimized by the back-propagation. The back-propagation adopts stochastic gradient descent to minimize the cross-entropy loss of the Att-DHSTNet. We also apply the dropout strategy to improve the capability of model generalization of our proposed model.

4. Performance Evaluation

In this section, we evaluate our proposed DHSTNet model. To show a comprehensive quantitative evaluation, we also compare our proposed model with other existing baselines. Finally, we apply an attention mechanism with our previous proposed method called as Att-DHSTNet, which shows the generalization of traffic flows prediction tasks.

Algorithm 1 Procedure: Att-DHSTNet Training Algorithm

```
Require: Historical observations: $\{X_t|t$=1,2,3,...n$\}$;
    External features: $\{E_t|t$=1,2,3,...n$\}$;
    The lengths of closeness, period influence and weekly influence are: $l_c$ ,$l_d$ ,$l_r$;
    Period influence span: d, Weekly influence span: r.
Ensure: Learned Att-DHSTNet Model.
D ← $\theta$;
// Modeling urban traffic values into spatio-temporal volumes.
for all available timestamps of training set t(2 ≤ t ≤ n) do
    $X_c$= $[X_{t-l_c}, X_t - (l_{c-1}), ..., X_{t-1}]$;
    $X_d$= $[X_{t-l_d \times d}, X_{t-(l_{d-1}) \times d}, ..., X_{t-1}]$;
    $X_r$= $[X_{t-l_r \times r}, X_{t-(l_{r-1}) \times r}, ..., X_{t-1}]$;
    // $X_t$ is the target at time t;
    Put a training instance ($X_c$, $X_d$, $X_r$, $E_t$) into D;
end for
//Training the model;
All learnable parameters initialize $\theta$ in Att-DHSTNet;
Repeat
Lr ← $h$
Lr=$e^{-\lambda t}$
select randomly a batch of samples $D_{batch}$ from D;
Feed each $X_c$, $X_d$, $X_r$, $E_t$ of an sample in $D_b$ into corresponding branch respectively;
Optimize $\theta$ by reducing the objective with $D_b$;
Until model converge
output learned Att-DHSTNet model
```

4.1. Experimental Settings and Datasets

In our experimental setup, we adopt the Linux server and other configuration are as follows:

- 8 Intel(R) Xeon(R) CPU E5-2680; v4 @2.40GHZ;.
- 4 NVIDIA P100 GPUs.
- RAM 256GB.
- cuDNN version 8.0, CUDA version 8.0.

In our experiment, we predict both inflow and outflow of citywide traffic flows on two public benchmarks, including TaxiBJ dataset for taxicab flow and BikeNYC dataset for bike flow prediction as shown in table 1. Each dataset contains flow trajectories and weather detailed as follows.

Table 1: Details of the evaluated datasets

	Datasets Statistics	
Dataset	**TaxiBJ**	**BikeNYC**
Type of data	GPS Taxi data	Bike rent
Location	Beijing City	New York City
Time period	01/07/2013-30/10/2013	01/04/2014-30/09/2014
	01/03/2014-30/06/2014	-
	01/03/2015-30/06/2015	-
	01/11/2015-10/04/2016	-
Time interval	half hour	1 hour
# Available time interval size	22,459	4,392
# Taxis/Bikes	34,000+	6,800+
Grid map size	32 × 32	16 × 8
Average sampling rate	about 60 seconds	-
Extrenal data (#holidays)	41	20
Weather conditions	16 types (e.g., cloudy, sunny)	-
Temperature/C°	[-24.6, 41.0]	-

- **TaxiBJ Dataset**: In this dataset, the taxicabs trajectories data are collected from Beijing for over 16 months: 01/07/2013-30/10/2013, 01/03/2014-

30/06/2014, 01/03/2015-30/06/2015, 01/11/2015-10/04/2016. The inflow and outflow are produced from more than 34000 trajectories taxis in Beijing. According to definition 2, we obtain two kinds of data, i.e.,, inflow and outflow of urban traffic crowd flows. In each dataset, we divide the data into two parts: (1) testing data and (2) training data. The testing data is the last four weeks' data, and the remained data as a training data. The external data includes weather conditions, temperature, and 41 categories of vacations.

- **BikeNYC Dataset**: The BikeNYC data generated from 01/04/2014-30/09/2014. The crowd flow maps of this dataset is 4,392 with a size of 16 × 8 and time interval is 1 hour. The bike-sharing data includes the duration of the trip, starting and ending station IDs, as well as initial and end times. In this dataset, the last 10 days are chosen as a testing set while all the remaining consider as a training set.

4.2. Compared Baselines

We compare our Att-DHSTNet model with widely used state-of-the-art following 10 baselines.

- **HA:** This model predicts the inflow and outflow of crowd flow through average historical flows in the corresponding periods.
- **ARIMA[13]:** This model used in the time series prediction problem.
- **LinUOTD[14]:** This model used for spatial and temporal information.
- **XGBoost [15]:** This model mainly used for boosting tree methods. The number of trees is set to 500, each maximum tree depth is 4 while 0.6 is the subsample rate.
- **Multilayer Perceptron (MLP):** We compare our model DHSTNet with a neural network that contains four fully connected layers.
- **ConvLSTM [16]:** This model adds convolutional layers to LSTM.

- **STDN [11]:** STDN used a deep hybrid neural network with an attention-based mechanism to model dynamic and spatio-temporal dependencies.

- **ST-ResNet [3]:** This method takes previous flows of data and applies the CNN model separately.

- **MST3D [17]:** This model apply 3D CNNs to jointly learn spatio-temporal correlation features from low-level to high-level layers for vehicle flows prediction.

- **DHSTNet[8]:** This model proposed in our previous paper. For TaxiBJ dataset, the number of residual units is 12 while for BikeNYC dataset is 4.

4.3. Implementation Details

Data Preprocessing: For TaxiBJ dataset, we divide the entire city map into 32 × 32 grid-based areas and set each period length for half an hour (30 minutes). Similarly, for BikeNYC, we divided the entire city map into 8 × 16 grid-based areas and set the length of each period to one hour. We scaled the traffic crowd flows value between [-1,1] by using the Min-Max normalization method. In the evaluation process, we re-scaled the predicted value back to as a normal value and compared with the ground truth. For external influence, we use the concept of one-hot coding schemes to transfer meta-data such as Weekend or Weekday, DayOfWeek, weather, and vacations into binary vectors, and use Min-Max normalization to scale the wind speed and temperature between [0, 1]. Similarly, we also apply Min-Max normalization to the existing baselines before compared them with our proposed DHSTNet model.

Hyperparameter Settings: To validate the experiments of our crowd flow prediction network, we adopt the PyTorch libraries with Tensorflow (version 1.2.1) and Keras (version 2.1.0). We set the inflow and outflow as a 2-channels flow in the generated volumes. Thus, the inflow and outflow of traffic flows can be predicted simultaneously. For BikeNYC implementation, the sequence lengths of three components (closeness, period influence, and weekly influence) are set to l_c=4, l_d=4, and l_r=4, respectively. Similarly, for TaxiBJ implementation, we set the three properties lengths (closeness, period influence, and weekly influence) to l_c=6, l_d=4, and l_r=4, respectively. The fully-connected layers and all convolutional layers are initialized by Xavier [18]. We used Batch normalization, and the minibatch size is set to 64 in the experiment. The learning rate (LR) is set to 0.001. An extra dropout layer is set to 0.25 dropout rate to reduce the over-fitting issue. To optimize our model, we use an end to end manner via an Adam optimization[19] by minimizing the Euclidean loss.

Performance Metrics: We adopt Root Mean Square Error (RMSE) and Mean Average Percentage Error (MAPE) as evaluation metrics. These two evaluation metrics are popular in crowd flows prediction that are extensively measure the performances of all methods. Particularly, these evaluation metrics are defined as follows:

$$RMSE = \sqrt{\frac{1}{z}\sum_{i=1}^{z}(\hat{x}_t - x_t)^2} \tag{10}$$

$$MAPE = \frac{1}{z}\sum_{i=1}^{z}\frac{|\hat{x}_t - x_t|}{x_t} \tag{11}$$

Where $\hat{x}_t$ and x_t denotes the predicted flow map and the real value map for time interval t, while z represent the number of samples used for validation.

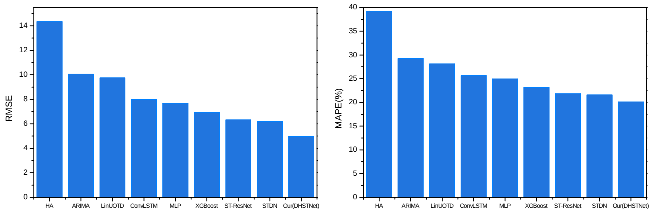
4.4. Performance Comparison and Result Analysis

The performance of the competing baselines and DHSTNet without attention for the inflow and outflow together on the BikeNYC and TaxiBJ is shown in figures 3 and 4 respectively. We can see that without attention our proposed DHSTNet model with four components, i.e.,, closeness, period influence, weekly influence and external components outperform the existing baselines by achieving the lowest RMSE and MAPE results on both datasets. For BikeNYC, the RMSE and MAPE values are 4.96 and 20.10%, respectively. Similarly, for TaxiBJ the values of RMSE and MAPE of our proposed DHSTNet are 15.25 and 14.20%, respectively.

Figure 5 and 6 present the results of existing methods and Att-DHSTNet on BikeNYC and TaxiBJ datasets with attention mechanism. The RMSE and MAPE values for BikeNYC dataset are 4.43 and 19.56% respectively. Likewise for TaxiBJ dataset, the RMSE and MAPE values are 14.28 and 13.56% respectively.

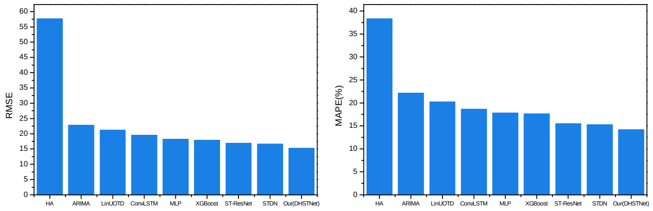
Table 2 shows the average results of our model for without attention and with attention mechanism as compared with existing baselines on BikeNYC and TaxiBJ datasets respectively. Table 3 and Table 4 present the detailed prediction results of inflow and outflow. We run each method 10 times and report the mean output of each method. From table 2 and 3, we can see that our DHSTNet model without attention mechanism achieves better performance than existing existing competing baselines. It proves that, the benefits of our proposed framework in describing spatio-temporal correlations of the urban traffic flows data.

After we combined with the spatio-temporal attention mechanism, our Att-DHSTNet model further reduces the predicting errors. It demonstrates that a good generalization of our model on the traffic flow prediction task. Noticeably, the traditional statistic time-series prediction methods (ARIMA and HA) cannot achieve better prediction accuracy. It reveals the weakness of methods of exclusively considering the relation of historical statistic values and overlooking the complex spatio-temporal dependency. The regression baselines (i.e.,, LinUOTD and XGBoost) extract spatial relationships as their characteristics. However, still, these models are unable to capture the dynamic spatial correlation and complex non-linear temporal dependencies. The model we proposed also produced better results than MLP and ST-ResNet. The important reason is that, MLP is hardly learned the linear mapping from historical data to the predicted output and insufficiently measured the spatio-temporal dependencies. The ST-ResNet used a three stacks of deep residual network to capture spatial dependencies along with three different periods. The results of convolutional are indiscriminately merged, which overlooks the typical impacts of short-term and long-term temporal dependencies. The ConvLSTM and STDN model showed a notable capability of modeling both the spatio-temporal dependencies by com-



(a) (b)

Figure 3: Evaluation results of BikeNYC without attention using (a) RMSE and (b) MAPE metrics [the MAPE values are %]



(a) (b)

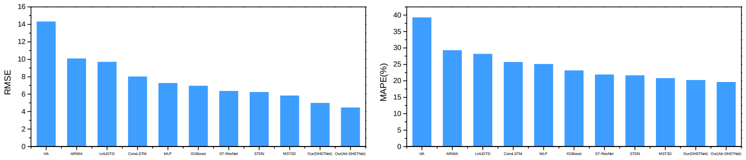
Figure 4: Evaluation results of TaxiBJ without attention using (a) RMSE and (b) MAPE metrics [the MAPE values are %]

bining both CNN and LSTM. However, the LSTM used limits their efficiencies on reaching long-term temporal dependencies. Apart from, independent modeling of spatio-temporal dependencies, it also limits their ability to capture complex spatio-temporal correlations. Given its ability of showing and attending to the entire spatio-temporal correlation, Att-DHSTNet has shown notable improvements compared to the previous deep learning baselines.

4.5. Effect of Four Components Without Attention Mechanism

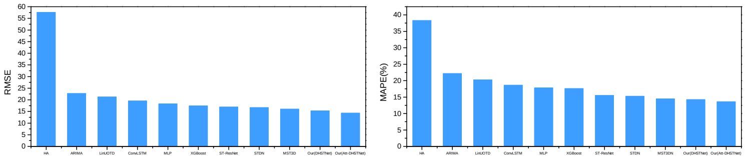
In this section, we describe the performance of four components, i.e.,, closeness, period influence, weekly influence, and external components, respectively by applying various variants of DHSTNet. The methods used in this study are listed as follows.

- **DHSTNet:** Our proposed method, which combines closeness, period influ-



(a) (b)

Figure 5: Evaluation results of BikeNYC with attention using (a) RMSE and (b) MAPE metrics [the MAPE values are %]



(a) (b)

Figure 6: Evaluation results of TaxiBJ with attention using (a) RMSE and (b) MAPE metrics [the MAPE values are %]

Table 2: Quantitative Performance Comparison On BikeNYC And TaxiBJ Datasets. We Run All Models 10 Times With Average Report

Baselines	**BikeNYC**		**TaxiBJ**	
	RMSE	MAPE	RMSE	MAPE
HA	14.29	39.19%	57.59	38.29%
ARIMA	10.06	29.21%	22.69	22.12%
LinUOTD	9.67	28.09%	21.19	20.21%
XGBoost	6.92	23.09%	17.39	17.59%
MLP	7.25	25.05%	18.25	17.83%
ConvLSTM	7.98	25.59%	19.52	18.63%
ST-ResNet	6.33	21.81%	16.89	15.48%
STDN	6.20	21.57%	16.65	15.27%
MST3D	5.81	20.68%	15.98	14.48%
Our(DHSTNet)	**4.96**	**20.10%**	**15.25**	**14.20%**
Our(Att-DHSTNet)	**4.43**	**19.56%**	**14.28**	**13.56%**

Table 3: Inflow and outflow results of different baselines for New York (BikeNYC) dataset.

Baselines	**inflow**		**outflow**	
	RMSE	MAPE	RMSE	MAPE
HA	14.03	38.86%	14.89	39.68%
ARIMA	9.96	28.96%	10.49	29.49%
LinUOTD	9.55	27.58%	10.11	28.74%
XGBoost	6.89	22.93%	7.06	23.43%
MLP	7.25	24.63%	8.07	25.29%
ConvLSTM	7.76	25.67%	8.33	25.73%
ST-ResNet	6.08	21.23%	6.63	22.17%
STDN	5.98	21.01%	6.51	21.96%
MST3D	5.66	20.21%	5.96	21.14%
Our(DHSTNet)	**4.78**	**19.60%**	**5.06**	**20.70%**
Our(Att-DHSTNet)	**4.28**	**19.57%**	**4.58**	**20.02%**

Table 4: Inflow and outflow results of different baselines for Beijing (TaxiBJ) dataset

Baselines	**inflow**		**outflow**	
	RMSE	MAPE	RMSE	MAPE
HA	57.46	37.69%	57.79	39.68%
ARIMA	22.57	22.13%	22.96	22.20%
LinUOTD	21.18	20.03%	21.54	20.43%
XGBoost	17.61	17.42%	18.23	17.69%
MLP	18.23	17.54%	18.30	18.21%
ConvLSTM	19.39	18.45%	19.94	18.69%
ST-ResNet	16.69	15.01%	17.01	17.78%
STDN	16.43	15.12%	16.78	15.44%
MST3D	15.98	14.71%	16.11	14.85%
Our(DHSTNet)	**15.24**	**13.80%**	**15.50**	**14.30%**
Our(Att-DHSTNet)	**14.27**	**13.72%**	**14.49**	**13.49%**

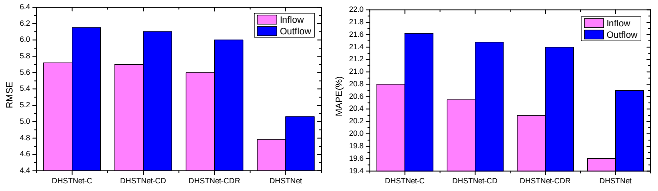
ence, weekly influence, and external branches, respectively.

- **DHSTNet-C:** This approach capture the spatio-temporal dependency of the closeness component.
- **DHSTNet-CD:** This approach includes both closeness and period branches.
- **DHSTNet-CDR:** This method uses three components, such as closeness, period influence, and weekly influence.

Figure 7 presents the results of our proposed DHSTNet without attention and its variants on the BikeNYC. Similarly, figure 8 present the results of TaxiBJ with its variants. We can see that the DHSTNet-C using only the closeness component has higher prediction accuracy than other methods. However, by adding the weekly influence and period influence, the performance is further improved. Again, considering multiple temporal dependencies can help improve the accuracy of traffic crowd flows prediction. An interesting observation is that, for the TaxiBJ dataset, the results of DHSTNet-C and DHSTNet-CD are similar (the RMSE of DHSTNet-CD is a bit higher than that of DHSTNet-C, while the MAPE of DHSTNet-CD is a bit lower than that of DHSTNet-C). It illustrates that adding a daily branch has a little effect on the TaxiBJ dataset. Therefore, it shows that the prediction accuracy of urban traffic crowd flows can be improved by considering multiple temporal correlations. The values of RMSE and MAPE can be further reduced by adding external components. It shows that external factors can significantly affect the performance of our method. Lastly, we can see that the method that combines the closeness, period influence, weekly influence, and the external branches can achieve the lowest RMSE and MAPE.

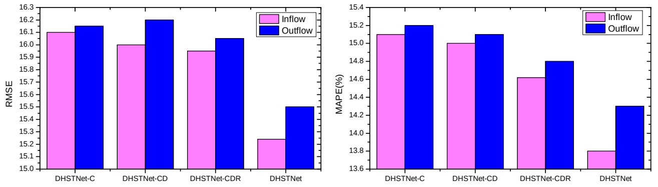
4.6. Results of Different Att-DHSTNet Variants

Here we empirically demonstrate the effectiveness of Att-DHSTNet with its variants results, including network depth, use of with external and without external data, and effect of filter number and filter size.



(a) (b)

Figure 7: Results of BikeNYC using (a) RMSE and (b) MAPE metrics [the MAPE values are %]



(a) (b)

Figure 8: Results of TaxiBJ using (a) RMSE and (b) MAPE metrics [the MAPE values are %]

4.6.1. Effect of Network Depth

Figure 9 shows the effect of network depth in more details for TaxiBJ. As to increase the number of residual units (i.e. goes deeper the network), first decreases the RMSE of the model and then the RMSE increases, it demonstrating that deeper the network has a better results. Because it not just to capture spatial but also capture distant dependency as well. However, when the residual units is ≥ 14 (i.e. the network is very deep and deep), the training process becomes more hard and there is possibility of overfitting becomes larger.

4.6.2. Effect Of External and Without External Data

Figure 10 shows the results of TaxiBJ and BikeNYC for external and without external data. We can see that, how the external factors can be useful to improve the prediction accuracy of our Att-DHSTNet model. As we discussed in section 2 that, the urban crowd flow is affected by these factors(special events, weather

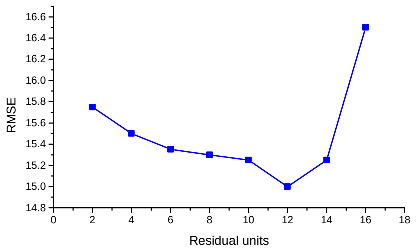
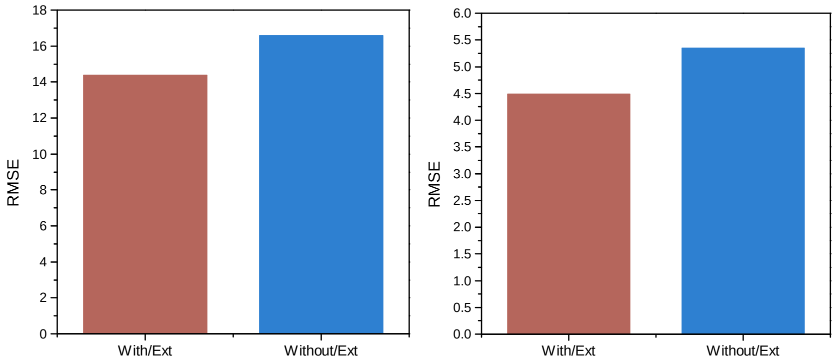


Figure 9: Effect of depth network



(a)

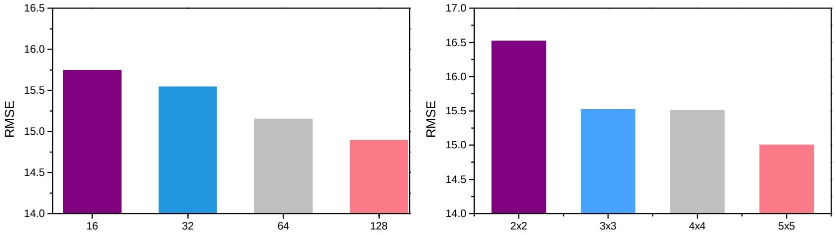
(b)

Figure 10: TaxiBJ and BikeNYC results with and without external data

etc).

4.6.3. Effect of Filter Numbers And Kernel Sizes

To get a better kernel size, we collected various combinations for training datasets. A right kernel size is important for better prediction accuracy. As shown in figure 11(b) how the kernel size affects the performance of our Att-DHSTNet model. Here we change the kernel sizes from 2 × 2 to 5 × 5. As we can see clearly in figure 11(b) larger the kernel size has lower the RMSE. It demonstrates that larger kernel size has better ability to depict the spatial dependency. Figure 11(a) shows that using more filter sizes has more better results.



(a)

(b)

Figure 11: Impact of various filter numbers and kernel sizes. We run 10 times each model

5. Related Work

Data-driven urban traffic crowd flows prediction problems have received large attention for decades. This section describes the related work of urban traffic crowd flows in more detail.

Deep Learning for Traffic Crowd Flow Prediction In recent years, many deep learning-based achievements have been made in the field of urban crowd flows prediction. The classical statistical models used for urban traffic crowd flow prediction which includes HA, VAR [20], Bayesian networks [21], Kalman filter [22], Markov chain [23] and ARIMA [13] etc. These models need the data to satisfy the same assumptions, but urban crowd flows data is too complex to satisfy these assumptions, so usually, they perform poorly in practice. Machine learning approaches such as SVM [24] and KNN [25] used to model more complex data, but they require careful feature engineering. Therefore, deep learning has brought about breakthroughs in many domains, including image processing and speech recognition. The other conventional deep learning methods are utilized. In [21] They exploit a Bayesian network to employ road adjacent links to analyze the impact of the data utilized for traffic predicting. Markov random method were proposed in [26] to identify the traffic congestion areas to solve the low resolution and uncertainty of geographical areas. A fuzzy Bayesian model in [27] for urban traffic prediction environment to extract spatio-temporal patterns. However, still these methods are failed to capture the dynamic and complex spatio-temporal correlations in the urban traffic flows data.

Over the last few decades, In the field of traffic flows prediction, deep learning-based approaches are becoming more and more popular due to the stronger expression capabilities of neural networks. In urban traffic prediction, two widely used aspects of the deep learning approaches i.e., CNN and LSTM. The CNN model has been successfully applied in many application areas, particularly in the area of computer vision [28], while the recurrent neural network (RNN) was applied for sequence-based learning [29]. The problem is that traditional RNN cannot give guarantee the retention of information long before. To predict the traffic conditions of different nodes[30], they proposed an stacked auto encoder (SAE) model. The research scholars proposed LSTM based framework [31], which significantly outperforms than traditional RNN. Currently, the LSTM network provides better results in sequence learning tasks such as speech recognition [32], machine translation [33] and text generation [34]. However, these existing models fail to comprehensively determine the temporal dependencies as well as ignore the complex correlations between spatio-temporal dependencies.

In urban crowd flows traffic prediction problem, we need to consider both spatio-temporal correlations, but neither CNN nor LSTM can dynamically implement these two spatio-temporal correlations. To address this problem, in [3], they proposed a residual-based framework named ST-ResNet to predict urban traffic flows. The ST-ResNet approach is much better than other time based-series methods. The ST-ResNet method is useful for spatio-temporal relations, while the problem is that this method uses simple combination of spatial features along with different time periods. It shows that the ST-ResNet model requires to manually characterize the temporal dependencies and cannot learned from the input data at the same time, which limits the ability of the model to flexibly and effectively capture the temporal dependency.

Attention-based Mechanism in Neural Networks Visual attention is a basic viewpoint of the human visual system, which alludes to the method by which people focus the computational assets of their brains visual system to particular regions of the visual field whereas perceiving the surrounding world. An attention-based mechanism has been successfully used in neural networks to handle various tasks such as vision question answering [35], natural language processing, image caption[36][37], machine translation[33] and speech recognition[38][39]. The main objective of attention-based mechanism is to identify information that is really hard from all input to the current task. In image caption task [36], they proposed two attention-based model and employ a visualization method to show the effect of attention-based mechanisms. To predict a time series[40], they proposed a multi-level attention model to adjust the correlations between multiple sensor time series. Similarly, for speech recognition[38], they proposed an RNN bi-directional and an attention mechanism for textual and vision question answering. A GRU-based attention mechanism were proposed in [35] for dynamic memory network. However, these methods time consuming since requires a well model to be trained for every time series. The main advantage of attention mechanism over traditional convolutional and LSTM structures are the network remember the previous output and put them all into consideration and also avoid overfitting.

In summary, motivated from the above research studies, we design a novel attention based mechanism with our previous DHSTNet model, called as Att-DHSTNet, which incorporates ConvLSTM units with an attention mechanism to learn the dynamic spatial and temporal correlations of urban traffic crowd flow data.

6. Conclusions

In this paper, we have proposed a novel deep hybrid spatio-temporal dynamic neural network called as DHSTNet for predicting citywide flow of crowds in each and every region of a city. The proposed DHSTNet model combines both CNN and LSTM, which takes the benefit of both spatio-temporal characteristics. We developed an attention-based system called as Att-DHSTNet, which simultaneously capture both spatial (near and far) and temporal (closeness, period influence, and weekly influence) dependencies as well as external factors (e.g., event, weather etc). It takes four properties into account i.e., closeness, period influence, weekly influence, and external component. Performance evaluation on two large scale real-world datasets show that the predicting accuracy of our proposed model is superior to existing state-of-the-art models.

In the future, our plan is to implement the training module on the cloud and then the prediction model is implemented on the edge computing or small datacenter, then the advantage is that each vehicle should quickly predict and take appropriate decisions for re-routing to avoid crowd or traffic flows congestion on the network.

References

[1] J. Zhang, F.-Y. Wang, K. Wang, W.-H. Lin, X. Xu, C. Chen, Data-driven intelligent transportation systems: A survey, IEEE Transactions on Intelligent Transportation Systems 12 (4) (2011) 1624–1639.

[2] S. H. Kim, J. Shi, A. Alfarrarjeh, D. Xu, Y. Tan, C. Shahabi, Real-time traffic video analysis using intel viewmont coprocessor, in: International Workshop on Databases in Networked Information Systems, Springer, 2013, pp. 150–160.

[3] J. Zhang, Y. Zheng, D. Qi, Deep spatio-temporal residual networks for citywide crowd flows prediction., in: AAAI, 2017, pp. 1655–1661.

[4] Y. Wang, Y. Zheng, Y. Xue, Travel time estimation of a path using sparse trajectories, in: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2014, pp. 25–34.

[5] 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.

[6] Z. Xu, Y. Wang, M. Long, J. Wang, M. KLiss, Predcnn: Predictive learning with cascade convolutions., in: IJCAI, 2018, pp. 2940–2947.

[7] J. Zhang, Y. Zheng, J. Sun, D. Qi, Flow prediction in spatio-temporal networks based on multitask deep learning, IEEE Transactions on Knowledge and Data Engineering (2019).

[8] A. Ali, Y. Zhu, Q. Chen, J. Yu, H. Cai, Leveraging spatio-temporal patterns for predicting citywide traffic crowd flows using deep hybrid neural networks, in: 2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS), IEEE, 2019, pp. 125–132.

[9] H. Yao, X. Tang, H. Wei, G. Zheng, Y. Yu, Z. Li, Modeling spatial-temporal dynamics for traffic prediction, arXiv preprint arXiv:1803.01254 (2018).

[10] 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.

[11] H. Yao, X. Tang, H. Wei, G. Zheng, Y. Yu, Z. Li, Modeling spatial-temporal dynamics for traffic prediction, arXiv preprint arXiv:1803.01254 (2018).

[12] X. Feng, J. Guo, B. Qin, T. Liu, Y. Liu, Effective deep memory networks for distant supervised relation extraction., in: IJCAI, 2017, pp. 4002–4008.

[13] B. M. Williams, L. A. Hoel, Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results, Journal of transportation engineering 129 (6) (2003) 664–672.

[14] 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.

[15] 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.

[16] 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, in: Advances in neural information processing systems, 2015, pp. 802–810.

[17] 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.

[18] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Proceedings of the thirteenth international conference on artificial intelligence and statistics, 2010, pp. 249–256.

[19] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).

[20] E. Zivot, J. Wang, Vector autoregressive models for multivariate time series, Modeling Financial Time Series with S-Plus® (2006) 385–429.

[21] S. Sun, C. Zhang, G. Yu, A bayesian network approach to traffic flow forecasting, IEEE Transactions on intelligent transportation systems 7 (1) (2006) 124–132.

[22] M. Lippi, M. Bertini, P. Frasconi, Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning, IEEE Transactions on Intelligent Transportation Systems 14 (2) (2013) 871–882.

[23] A. Abadi, T. Rajabioun, P. A. Ioannou, et al., Traffic flow prediction for road transportation networks with limited traffic data., IEEE Trans. Intelligent Transportation Systems 16 (2) (2015) 653–662.

[24] Y.-S. Jeong, Y.-J. Byon, M. M. Castro-Neto, S. M. Easa, Supervised weighting-online learning algorithm for short-term traffic flow prediction, IEEE Transactions on Intelligent Transportation Systems 14 (4) (2013) 1700–1707.

[25] J. Van Lint, C. Van Hinsbergen, Short-term traffic and travel time prediction models, Artificial Intelligence Applications to Critical Transportation Issues 22 (1) (2012) 22–41.

[26] P.-T. Chen, F. Chen, Z. Qian, Road traffic congestion monitoring in social media with hinge-loss markov random fields, in: 2014 IEEE International Conference on Data Mining, IEEE, 2014, pp. 80–89.

[27] M. Das, S. K. Ghosh, Fb-step: a fuzzy bayesian network based data-driven framework for spatio-temporal prediction of climatological time series data, Expert Systems with Applications 117 (2019) 211–227.

[28] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in neural information processing systems, 2012, pp. 1097–1105.

[29] R. J. Williams, D. Zipser, A learning algorithm for continually running fully recurrent neural networks, Neural computation 1 (2) (1989) 270–280.

[30] Y. Lv, Y. Duan, W. Kang, Z. Li, F.-Y. Wang, Traffic flow prediction with big data: a deep learning approach, IEEE Transactions on Intelligent Transportation Systems 16 (2) (2014) 865–873.

[31] F. Altché, A. de La Fortelle, An lstm network for highway trajectory prediction, in: 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), IEEE, 2017, pp. 353–359.

[32] O. Vinyals, S. V. Ravuri, D. Povey, Revisiting recurrent neural networks for robust asr, in: 2012 IEEE international conference on acoustics, speech and signal processing (ICASSP), IEEE, 2012, pp. 4085–4088.

[33] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, arXiv preprint arXiv:1409.0473 (2014).

[34] I. Sutskever, J. Martens, G. E. Hinton, Generating text with recurrent neural networks, in: Proceedings of the 28th International Conference on Machine Learning (ICML-11), 2011, pp. 1017–1024.

[35] C. Xiong, S. Merity, R. Socher, Dynamic memory networks for visual and textual question answering, in: International conference on machine learning, 2016, pp. 2397–2406.

[36] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, Y. Bengio, Show, attend and tell: Neural image caption generation with visual attention, in: International conference on machine learning, 2015, pp. 2048–2057.

[37] Z. Yang, Y. Yuan, Y. Wu, W. W. Cohen, R. R. Salakhutdinov, Review networks for caption generation, in: Advances in Neural Information Processing Systems, 2016, pp. 2361–2369.

[38] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, Y. Bengio, End-to-end attention-based large vocabulary speech recognition, in: 2016 IEEE international conference on acoustics, speech and signal processing (ICASSP), IEEE, 2016, pp. 4945–4949.

[39] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, Y. Bengio, Attention-based models for speech recognition, in: Advances in neural information processing systems, 2015, pp. 577–585.

[40] Y. Liang, S. Ke, J. Zhang, X. Yi, Y. Zheng, Geoman: Multi-level attention networks for geo-sensory time series prediction., in: IJCAI, 2018, pp. 3428–3434.