DeepSTN+: Context-aware Spatial-Temporal Neural Network for Crowd Flow Prediction in Metropolis

Ziqian Lin*, **Jie Feng***, **Ziyang Lu**, **Yong Li**†, **Depeng Jin**
Beijing National Research Center for Information Science and Technology
Department of Electronic Engineering, Tsinghua University, Beijing, China
{linzq14,feng-j16,zy-lu15}@mails.tsinghua.edu.cn,{liyong07,jindp}@tsinghua.edu.cn

Abstract

Crowd flow prediction is of great importance in a wide range of applications from urban planning, traffic control to public safety. It aims to predict the *inflow* (the traffic of crowds entering a region in a given time interval) and *outflow* (the traffic of crowds leaving a region for other places) of each region in the city with knowing the historical flow data. In this paper, we propose DeepSTN+, a deep learning-based convolutional model, to predict crowd flows in the metropolis. First, DeepSTN+ employs the *ConvPlus* structure to model the long-range spatial dependence among crowd flows in different regions. Further, PoI distributions and time factor are combined to express the effect of location attributes to introduce prior knowledge of the crowd movements. Finally, we propose an effective fusion mechanism to stabilize the training process, which further improves the performance. Extensive experimental results based on two real-life datasets demonstrate the superiority of our model, *i.e.*, DeepSTN+ reduces the error of the crowd flow prediction by approximately 8%∼13% compared with the state-of-the-art baselines.

Introduction

Spatial-temporal prediction is of great importance in a wide range of applications from urban planning, traffic control to public safety. In these applications, the government needs to forecast the crowd flows in the Eve celebrations to avoid potential catastrophic stampede; ride-sharing platforms like Uber are able to predict the travel demand around the city to provide better service for both consumers and drivers. In this paper, we study one of the classic problems of spatial-temporal prediction: *crowd flow prediction*.

As Figure 1 presents, crowd flow prediction (Zhang et al. 2016) is to predict the *inflow* (the total traffic of crowds entering a region in a given interval) and *outflow* (the total traffic of crowds leaving a region for other places during a given time interval) of each region in the city with knowing the historical flow information. Recently, to address this problem, deep learning-based models (Zhang et al. 2016; Zhang, Zheng, and Qi 2017; Zonoozi et al. 2018) are proposed, which achieve promising performance. Deep-ST (Zhang et al. 2016) is the first model to use a convolutional network to capture the spatial relations. Further, ST-ResNet (Zhang, Zheng, and Qi 2017) is proposed by replacing the general convolution operation with an advanced residual framework. By combining the pyramidal ConvGRU model with periodic representations, Periodic-CRN (Zonoozi et al. 2018) is designed to model the periodic nature of crowd flows explicitly.

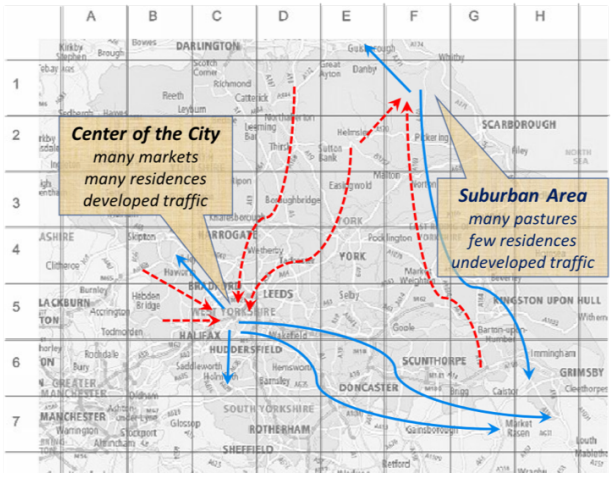


Figure 1: The dotted red line indicates inflow while the solid blue line indicates outflow. The curve line illustrates it's a distant movement while the straight line represents it's a movement of close range. $(5, C)$ is the center of the city while $(1, F)$ locates in the suburban area.

Nevertheless, existing approaches are still inefficient and inaccurate in practice due to the following three shortcomings:

1) *Failing to capture long-range spatial dependence among regions*. Due to the advanced transportation systems in the modern cities, people can quickly go anywhere in a short time by subway or taxi. Hence, long-range spatial relations between regions play an increasingly important role in crowd movements. Existing works (Zhang et al. 2016; Zhang, Zheng, and Qi 2017) use multi-layers convolutional network to model them. However, they can only capture the neighbor spatial dependence step by step, but fail to capture

*Equal contribution.
†Corresponding author.

long-range spatial dependence directly.

2) *Ignoring the influence of location function on the crowd movements*. Crowd mobility takes place in the physical world, which is distinctly influenced by the attributes of the location (Xu, Zhang, and Li 2016). For example, people usually go to office from home in the morning and come back in the evening. Obviously, the attributes, more precisely, the function of location contains some prior knowledge about the human movement and crowd mobility. However, none of the existing solutions take the characteristics of location into consideration.

3) *Redundant and unstable neural network structure.* ST-ResNet (Zhang, Zheng, and Qi 2017) utilizes three independent branches of residual convolutional units to process different inputs and directly fuse them with a linear operation at the end of the model. However, the end-fusion mechanism results in the deficiency of interaction between various components which also leads to inefficient parameters and unstable characteristics of the network.

In summary, long-range spatial dependence, the effect of location and more effective fusion mechanism should be taken into consideration. In this paper, we propose DeepSTN+ by specially designing structures to address these challenges mentioned above. First, we design a *ConvPlus* structure to capture the long-range spatial dependence between the crowd flows directly. *ConvPlus* is placed in the head of a general residual unit as a global feature extractor to extract global dependence among regions. Second, we design a *SemanticPlus* structure to learn the prior knowledge of location on crowd movements. With the static geographical distributions of PoI (point of interest) as input, *SemanticPlus* utilizes the time factor to give different weights to different PoIs at different times. Finally, we introduce early-fusion and multi-scale fusion mechanism in DeepSTN+ to reduce the trainable parameters and capture complicated relations between features of different levels. In this way, our system is able to model more complicated spatial correlations to achieve better performances. Our contributions can be summarized as follows:

- We design a new residual unit, the ResPlus unit to replace the original residual unit in ST-ResNet. We point out that ordinary convolutional models cannot extract long-range spatial dependence effectively. The proposed ResPlus unit contains a *ConvPlus* structure which is able to capture long-range spatial dependence between crowd flows.
- We design a *SemanticPlus* structure to model different effects of different locations to learn the prior knowledge of the crowd movements. And we apply early-fusion mechanism in the head of DeepSTN+ and multi-scale fusion mechanism at the end of DeepSTN+ to improve both the accuracy of prediction and the stability of the model.
- We conduct extensive experiments based on two real-life mobility datasets with 5 baselines including the state-of-the-art model ST-ResNet. Compared with the state-of-the-art approach, results demonstrate that our model can reduce the error of crowd flow prediction by about 8%$\sim$13%.

Preliminaries

In this section, we first formally introduce the crowd flow prediction problem, and then briefly review ST-ResNet (Zhang, Zheng, and Qi 2017) as background knowledge.

Problem Formulation

Definition 1 (Region (Zhang et al. 2016)) To indicate the regions of the city, we partition a city into an $H \times W$ grids based on the longitude and latitude, where all grids have the same size and each grid represents a region.

Definition 2 (Inflow/outflow (Zhang et al. 2016)) To express the crowd flows in the city, we define inflow and outflow for the region (h, w) at the i^{th} time interval as follows:

$$x_i^{h,w,in} = \sum_{Tr_k \in \mathbb{P}} |\{j > 1 | g_{j-1} \notin (h,w) \ \ \& \ \ g_j \in (h,w)\}|,$$

$$x_i^{h,w,out} = \sum_{Tr_k \in \mathbb{P}} |\{j \geq 1 | g_{j-1} \in (h,w) \ \ \& \ \ g_j \notin (h,w)\}|.$$

Here the $\mathbb{P}$ represents the collection of trajectories at the i^{th} time interval. $Tr : g_1 \rightarrow g_2 \rightarrow \cdots \rightarrow g_{|Tr|}$ is a trajectory in $\mathbb{P}$, and g_j is the geospatial coordinate; $g_j \in (h, w)$ means the point g_j lies within grid (h, w), and vice versa; $|\cdot|$ denotes the cardinality of a set.

Crowd Flow Prediction: Given the historical observations $\{\mathbf{X}_i | i = 1, 2, \cdots, n-1\}$, predict $\mathbf{X}_n$.

ST-ResNet (deep spatio-temporal residual networks) contains four major components: *closeness*, *period*, *trend* and external unit. Each component leads to a predicted crowd flow map through a branch of residual units or a fully-connected layer. Then, the model uses an end-fusion which is a linear combination to fuse all these predictions. The external factors of ST-ResNet contain weather, holiday event, and metadata.

Convolutional neural network (CNN) has shown great power to hierarchically capture regional features in many pictures (Simonyan and Zisserman 2014; Szegedy et al. 2015), and kernels of these convolutions usually have a small size, which means they cannot capture the relationships in long-distance range directly (kernel size is 3x3 in the ST-ResNet). However, the long-range spatial dependence of crowd flows is increasingly significant with the development of traffic in the city. On the other hand, ST-ResNet ignores the effect of location on the crowd movements. Moreover, the end-fusion mechanism of ST-ResNet leads to the deficiency of interaction, the inefficiency of parameters as well as unstable characteristics of the model.

Our Model

Figure 2 shows the framework of our model. It mainly consists of three components: flow input, SemanticPlus, and ResPlus units. Flow input contains *closeness*, *period* and *trend* and can be reduced to *closeness* and *period* due to the the limitation of the time range of the data. SemanticPlus contains PoI distributions and time information. ResPlus units can capture long-range spatial dependence. Inflow and outflow in each region are counted every hour or every half an

hour to form series of crowd flow maps. These flow maps (population distribution maps) are Min-Max normalized to $[-1, 1]$. As illustrated in Figure 2, population distribution maps are selected corresponding to recent time, near history and distant history for the input of the network. Different categories of PoI distributions are Min-Max normalized to $[0, 1]$. As shown in the top-left part of Figure 2, PoI distribution maps are weighted by the time information separately. After that, PoI information and crowd flow information are early-fused and then fed to the stack of ResPlus units. Finally, features of different levels of ResPlus units are fused together into a convolution part, and then mapped into $[-1, 1]$ by a Tanh function (LeCun et al. 2012). The details of ResPlus, SemanticPlus and fusion mechanism will be introduced below:

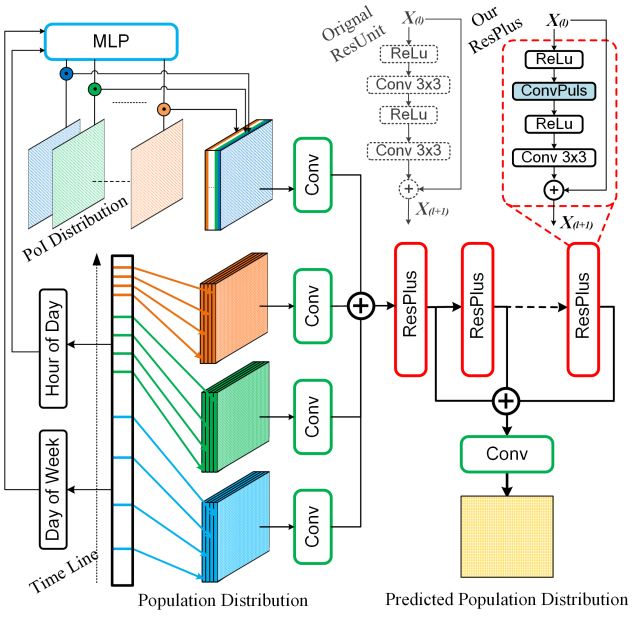


Figure 2: DeepSTN+ architecture, where Conv is Convolution; ResPlus is ResPlus Unit; MLP is Multi-Layer Perceptron.

ResPlus

Many deep learning-based models were proposed for crowd flow prediction mainly including two basic structures: RNN-based structure like ConvLSTM (Xingjian et al. 2015) and Periodic-CRN (Zonoozi et al. 2018) and CNN-based structure such as Deep-ST (Zhang et al. 2016) and ST-ResNet (Zhang, Zheng, and Qi 2017). However, the training of RNN-based structures always consumes a lot of time. Hence, we choose the CNN-based structure ST-ResNet as our basic model.

Convolutional neural network (CNN) has shown its powerful ability to capture the regional features of pictures (Krizhevsky, Sutskever, and Hinton 2012; Simonyan and Zisserman 2014; Szegedy et al. 2015). However, with the development of transportation systems in modern cities, long-range spatial dependence is increasingly important. Therefore, in this paper, we design ConvPlus to capture long-range spatial dependence of crowd flows in the city. As shown in Figure 3, the ResPlus unit employs a ConvPlus and an ordinary convolution. We also attempt Batch Normalization (BN) (Ioffe and Szegedy 2015) and Dropout (Srivastava et al. 2014) in the designed ResPlus unit, which are not shown in the figure for convenience.

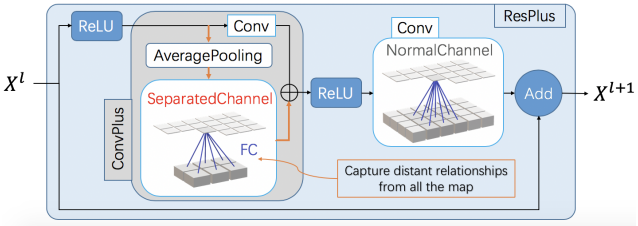


Figure 3: The architecture of ResPlus unit, where FC denotes a fully-connected layer. An ordinary Conv (right) and a ConvPlus (left) are shown in this figure. Normal channels capture close relationships while separated channels capture long-range spatial dependence. An average pooling layer is used to reduce the number of parameters.

Each channel of an ordinary convolution corresponds to a kernel. The convolution uses these kernels to calculate cross-correlation coefficients with the map, *i.e.*, capture the features of the map. Kernels of an ordinary convolution usually have the same kernel size which is much smaller than the size of a crowd flow map. In ST-ResNet and DeepSTN+, kernels of convolutions typically have the size 3x3. However, long-distance relationships of crowd flows widely exist in cities. For instance, some people go to work by subway for a long distance. We call this kind of relationships *long-range spatial dependence*. Long-range spatial dependence varies from place to place, which makes it difficult for a stack of convolutions to capture this relationship effectively.

As shown in the left part of Figure 3, in the ConvPlus structure we separate some channels of an ordinary convolution to capture long-range spatial dependences of each region. A fully-connected layer is used to capture long-range spatial dependence directly between every pair of regions, and an average pooling layer is set before this layer to reduce the number of parameters. Hence, there are two kinds of channels in the output of ConvPlus. The output of ConvPlus has the same shape as a normal convolution output and can be used as the input of the next convolution.

Figure 4 above shows two heatmaps of spatial dependence for two different regions represented by red and yellow stars. These target regions have not only regional dependence, but also long-range spatial dependence with some distant regions. It is also shown different regions have different relationships with all the map, which is difficult for stacks of ordinary convolutions to capture effectively.

For the reason that the output of ConvPlus has two different kinds of channels, we use ConvPlus+Conv rather than ConvPlus+ConvPlus in the ResPlus unit. DeepSTN+ with-

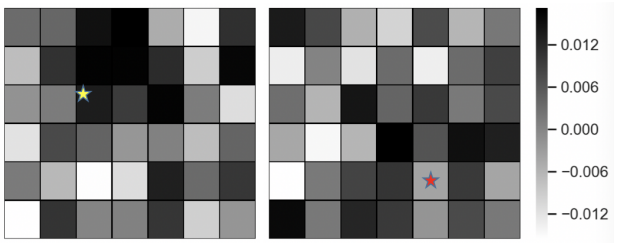


Figure 4: Heatmaps of two different regions with 3x3 pooling size.

out SemanticPlus is formulated as below:

$$\widehat{\mathbf{X}} = f_{Res}(f_{EF}(\mathbf{X}^c + \mathbf{X}^p + \mathbf{X}^t)),$$

where $\mathbf{X}^c$,$\mathbf{X}^p$ and $\mathbf{X}^t$ denote three types of historical crowd flow maps–*closeness*, *period* and *trend*. $\widehat{\mathbf{X}}$ denotes the predicted crowd flow map. The notation $+$ indicates the concatenate operation. The function f_{EF} indicates a convolution used to early-fuse different kinds of information and Function f_{Res} suggests a stack of ResPlus units.

SemanticPlus

PoI has significant influences (Xu, Zhang, and Li 2016; Cheng et al. 2013; Gao et al. 2015) on human mobility, and these influences vary from time to time (Yuan et al. 2013). Thus, we integrate his prior knowledge into our model. We collect PoI information including type, amount and location. Then, we count the number of PoI in each grid and use a 1-channel matrix to denote each kind of PoI distribution. Figure 5 shows the flow distribution map and the food distribution map of Beijing City. Their distributions are similar, and the cross-correlation coefficient of them is 0.87 implying their potential associations.

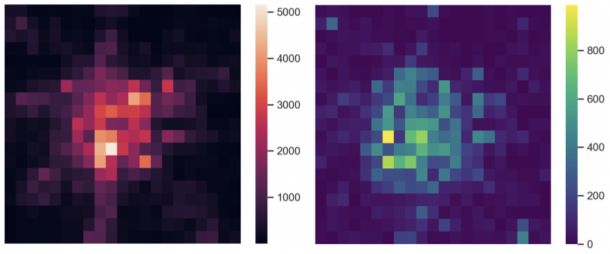


Figure 5: Examples of a distribution map of crowd flow (left) and a distribution map of food (right).

We use a time-vector to indicate the time of each crowd flow map. Time-vector consists of two parts: a 1-hot vector is used to indicate the time in a day, whose length is 24 if the time interval of flow maps is one hour; another 1-hot vector indicates the day in a week, whose length is 7. A time-vector is the combination of these two 1-hot vectors.

To model that PoI has varied temporal influences on flow maps, we transform the time-vector to the influence strength of PoI. We use tensor $\mathbf{X}^s$ of the size $PN \times H \times W$ to indicate the PoI maps (PN indicates the number of PoI categories. H and W are the height and width of the grid map), a vector $\mathbf{I}$ to indicate the time-vector, and a vector $\mathbf{R}$ of the size PN to indicate the influence strength of PoI. Thus, we have the time-weighted PoI distributions, formulated as below:

$$\mathbf{S} = \mathbf{X}^s * \mathbf{R} = \mathbf{X}^s * f_t(\mathbf{I}),$$

where function $f_t()$ transforms the time-vector to the influence strength of the PoI. The notation $*$ means each PoI distribution map will be weighted by a number, which is the influence strength of PoI. We assume that PoI in different regions of the same category has the same time pattern. Therefore, the PoI distribution map of a single category is weighted by the same number. Figure 6 shows the influence strength of recreation and residence. The influence strength varies from time to time in a week, and some regular patterns exist in every day. Many people go to work in the morning and return home after work, so there are two obvious peaks every day in the morning and afternoon for residence. Compared with residence, the influence of recreation on crowd flows is relatively stable.

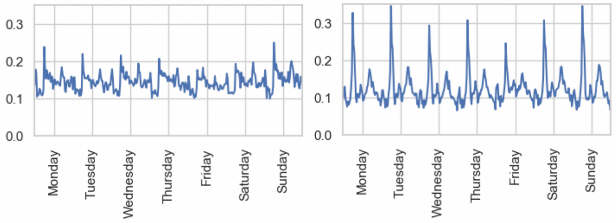


Figure 6: The influence strength of recreation and residence.

Fusion

Instead of linear combination, more complex interactions should exist in *closeness*, *period* and *trend*. These flow information also has complicated interactions with PoI distributions. To model these interactions, we use early-fusion instead of end-fusion to make different kinds of information interact with each other earlier. Early-fusion also reduces the number of parameters by almost 2/3 compared with the end-fusion of ST-ResNet. Moreover, ST-ResNet is occasionally unable to converge. We find that this problem can be solved by early-fusion for the reduction of parameters and simplification of the network. Considering the features of different layers have different functions, we set a multi-scale fusion mechanism at the end of the network (shown in the below-right part of Figure 2). Here we formulate the whole network as below:

$$\widehat{\mathbf{X}} = f_{con}(f_{Res}(f_{EF}(\mathbf{X}^c + \mathbf{X}^p + \mathbf{X}^t + \mathbf{S}))),$$

where the function f_{EF} suggests a convolution to early-fuse different kinds of information, which compress the number of channels before early-fusion for the input of subsequent ResPlus units. The function f_{con} suggests the ultimate multi-scale fusion, which means a concatenate layer followed by a convolution layer. The notation $\mathbf{S}$ indicates the output of SemanticPlus, *i.e.*, the time-weighted PoI distributions.

Training

Algorithm 1 outlines the training procedure for DeepSTN+. We construct training instances from the original series of crowd flow maps and PoI information (lines 1-7), including $\mathbf{X}_i^c, \mathbf{X}_i^p, \mathbf{X}_i^t$ as three types of historical crowd flows and $\mathbf{X}^s, \mathbf{I}_i$ as the inputs of SemanticPlus. All of these inputs vary from time to time except the PoI information $\mathbf{X}^s$. DeepSTN+ is trained via back-propagation and Adam (Kingma and Ba 2014) (lines 8-12).

Procedure: DeepSTN+ Training Procedure

Input: historical observations: $\{\mathbf{X}_0, \cdots, \mathbf{X}_{n-1}\}$;
PoI distributions: $\mathbf{X}^s$; time-vector:$\{\mathbf{I}_0, \cdots, \mathbf{I}_{n-1}\}$;
length of closeness, period, trend sequences: lc, lp, lt;
period span: p; trend span: t.
Output: Learned DeepSTN+ model

// construct the training data $\mathbb{D}$
1 $\mathbb{D} \Leftarrow \emptyset$
2 **for** all available time interval:
3 $\quad \mathbf{X}_i^c = [\mathbf{X}_{i-l_c}, \mathbf{X}_{i-(l_c-1)}, \cdots, \mathbf{X}_{i-1}]$
4 $\quad \mathbf{X}_i^p = [\mathbf{X}_{i-l_p \cdot p}, \mathbf{X}_{i-(l_p-1)\cdot p}, \cdots, \mathbf{X}_{i-p}]$
5 $\quad \mathbf{X}_i^t = [\mathbf{X}_{i-l_t \cdot t}, \mathbf{X}_{i-(l_t-1)\cdot t}, \cdots, \mathbf{X}_{i-t}]$
6 $\quad$ put an training instance $(\{\mathbf{X}_i^c, \mathbf{X}_i^p, \mathbf{X}_i^t, \mathbf{X^s}, \mathbf{I}_i\}, \mathbf{X}_i)$ into $\mathbb{D}$
7 **end** // $\mathbf{X}_i$ is the target at time i
// train the model
8 initialize all learnable parameters θ in DeepSTN+
9 **repeat**
10 $\quad$ randomly select a batch of instances $\boldsymbol{D}$ from $\mathbb{D}$
11 $\quad$ optimize θ using Adam and $\boldsymbol{D}$
12 **until** model overfitting

Performance Evaluation

In this section, we conduct extensive experiments based on two real-world datasets with different types of flows in different cities to answer the following three research questions:

- RQ1: Does our proposed DeepSTN+ outperform existing algorithms in crowd flow prediction?
- RQ2: How do ResPlus, SemanticPlus, and early-fusion improve the performance of DeepSTN+ on crowd flow prediction task?
- RQ3: How do the hyper-parameters of DeepSTN+ effect the performance of prediction task?

Datasets

Two datasets showing in Table 1 are used in our experiments. Each dataset contains two sub-datasets: flow trajectories and PoI information.

***MobileBJ*:** This dataset is collected from the most popular social network vendor in China from Apr. 1st to Apr. 30th. It records the locations of users whenever they request the location service in the application. We transform them into grid maps of crowd flows as **Definition 2**. We choose data from the last week as the testing data, and all data before that as the training data. Table 2 shows 17 categories of PoIs in this dataset.

***BikeNYC*:** This dataset is taken from the NYC Bike system in 2014, from Apr. 1st to Sept. 30th. Trip data includes trip duration, starting and ending station IDs, and start and end times. Among the data, the last 14 days are chosen as testing data, and the others as training data. We collect 9 types of PoIs for this dataset as shown in Table 2.

Dataset	MobileBJ	BikeNYC
Data type	Mobile application	Bike rent
Location	Beijing	New York
Time span	4/1/2018-4/30/2018	4/1/2014-9/30/2014
Time interval	30 minutes	1 hour
Grid map size	(19,21)	(21,12)
PoI Num	264581	26202

Table 1: Datasets

Dataset	Point of Interests (PoI)
BikeNYC	Food, Residence, ShopServic, CollegeUniversity, NightlifeSpot, TravelTransport, ArtEntertainment, ProfessionalOtherPlace, OutdoorsRecreation
MobileBJ	Food, Hotel, Culture, Sports, Shopping, Factory, Recreation, Institution, MedicalCare, ScenicSpot, Education, Landmark, Residence, TravelTransport, BusinessAffairs, LifeService

Table 2: categories of PoIs for BikeNYC and MobileBJ

Baselines

We compare our DeepSTN+ model with the following 5 baselines:

- **HA**: It predicts inflow and outflow of crowds by the average value of historical inflow and outflow in the corresponding periods.
- **VAR** (Hamilton 1994): Vector Auto-Regressive can capture the pairwise relationships among all flows but has massive computational costs due to a large number of parameters.
- **ARIMA** (Box et al. 2015): Auto-Regressive Integrated Moving Average is a combination of AR (autoregression) and MA (moving average) with difference process.
- **ConvLSTM** (Xingjian et al. 2015): It is a neural network (combination of convolution and LSTM) capturing both spatial and temporal features but consuming a lot of time to training due to the recurrent structure.
- **ST-ResNet** (Zhang, Zheng, and Qi 2017): It's a CNN-based model for spatial-temporal data, which shows state-of-the-art results on crowd flow prediction.

Metrics and Parameters

We use Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) as metrics:

$$RMSE = \sqrt{\frac{1}{T}\sum_{i=1}^{T} ||\mathbf{X}_i - \widehat{\mathbf{X}}_i||_2^2},$$

$$MAE = \frac{1}{T}\sum_{i=1}^{T}|\mathbf{X}_i - \widehat{\mathbf{X}}_i|,$$

where $\mathbf{X}_i$ and $\widehat{\mathbf{X}}_i$ denote the ground-truth and the prediction at the i^{th} time interval. T is the total number of samples in the testing data. RMSE is also used as the loss function of DeepSTN+.

Parameter	**BikeNYC**	**MobileBJ**
All channels in ConvPlus	64	64
Separated channels in ConvPlus	8	8
Categories of PoIs	9	16
Number of ResPlus units	2	2
Number of flow maps in a day	24	48
Pooling rate	1	3

Table 3: Settings of parameters.

Table 3 shows the settings of different parameters. To compare with ST-ResNet, the number of channels of Conv and ConvPlus in DeepSTN+ is 64, the same as ST-ResNet. ST-ResNet has three branches of residual convolutional units for *closeness*, *period* and *trend* respectively, while our DeepSTN+ only has one branch of ResPlus units. For the reason that ResPlus unit is able to capture long-range spatial dependence, the number of ResPlus units as 2 in DeepSTN+ model, while each branch of ST-ResNet has 4 residual units. We find that in both datasets the model works best when the number of separated channels in ConvPlus is 8. Considering that the dataset BikeNYC is sparser and more discontinuous than MobileBJ in spatial measurement, the pooling rate of the average the pooling layer is 1 for BikeNYC and 3 for MobileBJ.

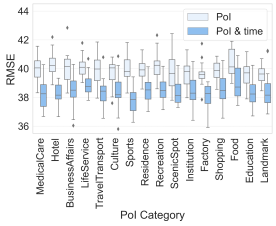
Model	RMSE	Δ	MAE
HA	136.32	223.14%	51.60
VAR	62.75	48.76%	44.27
ARIMA	58.63	28.43%	30.05
ConvLSTM	44.31	5.04%	27.75
ST-ResNet	42.19	0	26.95
DeepSTN	39.85	-5.54%	26.53
DeepSTN+plus	37.69	-10.67%	23.85
DeepSTN+PoI	39.12	-7.27%	25.87
DeepSTN+PoI∗time	37.62	-10.83%	24.89
DeepSTN+plus+PoI∗time	**36.29**	**-13.97%**	**22.94**
DeepSTN+plus+PoI∗time+con	**36.89**	**-12.56%**	**23.43**

Table 4: Comparison among different baselines and variants of DeepSTN+ on MobileBJ.

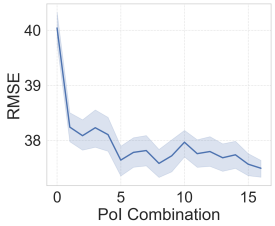
Performance comparison Table 4 and table 5 show the performances of baselines and variants of our model. The notation Δ indicates the reduction of error compared with ST-ResNet. DeepSTN suggests the model employs the early-fusion mechanism to fit complex interactions among different information but using ordinary residual convolutional units without ResPlus structure and PoI information, which brings about 4%∼5% improvement. +plus indicates the model employs the ResPlus units to capture long-range spatial dependence, which brings another 3% improvement on average. +PoI indicates the model employs PoI but without time information. +PoI*time indicates the model employs both PoI and time information. With time information, PoI reduces the error of the model by 7%∼10%. +con indicates the ultimate multi-scale fusion. However, multi-scale fusion mechanism only works on BikeNYC due to different properties of these two datasets. Most importantly, when employing both ResPlus units and PoI with the time information, the model obtains 8%∼13% enhancement based on two real-life datasets, which shows the superiority of our model.

Model	RMSE	Δ	MAE
HA	7.885	21.79%	2.823
VAR	10.097	55.94%	5.49
ARIMA	10.894	68.25%	3.246
ConvLSTM	6.412	-0.97%	2.543
ST-ResNet	6.475	0	2.395
DeepSTN	6.213	-4.05%	2.388
DeepSTN+plus	6.128	-5.36%	2.362
DeepSTN+PoI	6.191	-4.39%	2.381
DeepSTN+PoI∗time	6.021	-7.01%	2.340
DeepSTN+plus+PoI∗time	**5.984**	**-7.58%**	**2.292**
DeepSTN+plus+PoI∗time+con	**5.955**	**-8.03%**	**2.285**

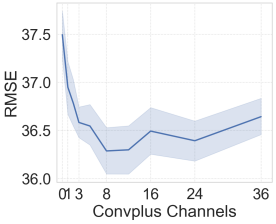
Table 5: Comparison among different baselines and variants of DeepSTN+ on BikeNYC.



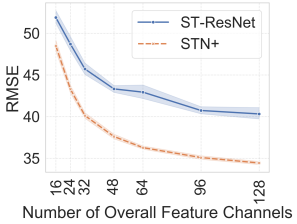
(a) Effect of PoI categories and time information.



(b) Effect of PoI combinations.



(c) Effect of separated channels.



(d) Effect of overall channels.

Figure 7: Effect of hyper-parameters

Effects of hyper-parameters Figure 7 shows the effects of hyper-parameters. Figure 7(a) shows the effect of PoI categories and time information, where the y-axis is the prediction error and the x-axis is the category of different PoIs. Results of each kind of PoI, with and without time information are presented, showing that the performances of PoIs vary

from category to category but are all improved by the time information. Figure 7(b) shows the effect of the PoI combinations, where the x-axis is the number of PoI categories we feed into DeepSTN+. The results show that with more prior knowledge, the model performs better. Figure 7(c) shows the effect of the number of separated channels for long-range spatial dependence. When the number of separated channels changes from 0 to 1, the model gets a sudden benefit, which means the long-range spatial dependence of the city is significant and captured by the ConvPlus structure. The model performs best when the number of separated channels is 8 and gets worse when adding more separated channels to ConvPlus, which suggests that both local relationships and long-range spatial dependence are meaningful. Figure 7(d) shows the effect of the number of all channels in the convolution and ConvPlus. We select the number of all channels ranging from 16 to 128. The results show our model outperforms ST-ResNet about 10% steadily.

In summary, time information improves the performance of PoI significantly. With more PoI information, the model performs better. DeepSTN+ captures both local relations and long-range spatial dependence to achieve good results.

RELATED WORK

We review some previous works on crowd flow prediction. STW-KNN (Xia et al. 2016) is an improved KNN (K-nearest neighbor algorithm) model to enhance forecasting accuracy based on spatio-temporal correlation. CityMomentum (Fan et al. 2015) uses a mixture of multiple random Markov chains, each of which is a naive movement predictive model. The seasonal and trend models (Hoang, Zheng, and Singh 2016) are built as intrinsic Gaussian Markov random fields, whereas a residual model exploits the spatio-temporal dependence among different flows and regions. Works mentioned above are based on traditional algorithms.

Recently, many deep learning-based models were proposed for crowd flow prediction. ConvLSTM (Xingjian et al. 2015), hybrid deep learning framework (Du et al. 2017), STRCNs (Jin et al. 2018) all explored the combination of convolution and LSTM. By combining the pyramidal ConvGRU model with periodic representations, PeriodicCRN (Zonoozi et al. 2018) was designed to model the periodic nature of crowd flow explicitly. These deep learning-based models mainly focus on the combination of CNN and RNN. However, the training of RNN-based structures always consumes a lot of time. Deep-ST (Zhang et al. 2016) is the first model to use a convolutional network to capture spatial relations. Further, ST-ResNet (Zhang, Zheng, and Qi 2017) was proposed by replacing the general convolution operation with an advanced residual framework. These two CNN-based networks employ convolutional stacks to capture spatial dependence while using *closeness*, *period* and *trend* as input to capture temporal dependence. All of these deep learning-based works have noticed the long-term temporal dependence of crowd mobility and tried to find better structures to express it. However, none of them constructed a special structure for long-range spatial dependence. Moreover, the effect of location attributes was also ignored in these works.

In this paper, we mainly construct a new structure for long-range spatial dependence and employ semantic information to express the effect of location attributes.

Conclusion

We propose DeepSTN+ for crowd flow prediction, which simulates long-range spatial dependence, considers the effect of location attributes, and employs an appropriate fusion mechanism. We conduct our experiments based on two types of real-life datasets, achieving performances which are significantly beyond the state-of-the-art model ST-ResNet and the other 4 baselines, confirming that our model is more applicable for crowd flow prediction. In the future, we will consider the combination of different types of datasets to cover crowd flows in the city more comprehensively (*e.g.*, flows of people, bikes and cars exist in the city at the same time). On the other hand, in order to express the long-range spatial dependence more precisely, we will also concentrate on the location-aware and time-aware attention mechanism for each region to achieve more precise perception (*e.g.*, the relationship between residence and workplace is strong on weekdays but weak at weekends).

Acknowledgments

This work was supported in part by The National Key Research and Development Program of China under grant 2017YFE0112300, the National Nature Science Foundation of China under 61861136003, 61621091 and 61673237, Beijing National Research Center for Information Science and Technology under 20031887521, and research fund of Tsinghua University - Tencent Joint Laboratory for Internet Innovation Technology.

References

Box, G. E.; Jenkins, G. M.; Reinsel, G. C.; and Ljung, G. M. 2015. *Time series analysis: forecasting and control*. John Wiley & Sons.

Cheng, C.; Yang, H.; Lyu, M. R.; and King, I. 2013. Where you like to go next: Successive point-of-interest recommendation. In *IJCAI*, volume 13, 2605–2611.

Du, S.; Li, T.; Gong, X.; Yang, Y.; and Horng, S. J. 2017. Traffic flow forecasting based on hybrid deep learning framework. In *Intelligent Systems and Knowledge Engineering (ISKE), 2017 12th International Conference on*, 1–6. IEEE.

Fan, Z.; Song, X.; Shibasaki, R.; and Adachi, R. 2015. Citymomentum: an online approach for crowd behavior prediction at a citywide level. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 559–569. ACM.

Gao, H.; Tang, J.; Hu, X.; and Liu, H. 2015. Content-aware point of interest recommendation on location-based social networks. In *AAAI*, 1721–1727.

Hamilton, J. D. 1994. *Time series analysis*, volume 2. Princeton university press Princeton, NJ.

Hoang, M. X.; Zheng, Y.; and Singh, A. K. 2016. Forecasting citywide crowd flows based on big data. *ACM SIGSPATIAL* 2016.

Ioffe, S., and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

Jin, W.; Lin, Y.; Wu, Z.; and Wan, H. 2018. Spatio-temporal recurrent convolutional networks for citywide short-term crowd flows prediction. In *Proceedings of the 2nd International Conference on Compute and Data Analysis*, 28–35. ACM.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 1097–1105.

LeCun, Y. A.; Bottou, L.; Orr, G. B.; and Müller, K.-R. 2012. Efficient backprop. In *Neural networks: Tricks of the trade*. Springer. 9–48.

Simonyan, K., and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research* 15(1):1929–1958.

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1–9.

Xia, D.; Wang, B.; Li, H.; Li, Y.; and Zhang, Z. 2016. A distributed spatial–temporal weighted model on mapreduce for short-term traffic flow forecasting. *Neurocomputing* 179:246–263.

Xingjian, S.; Chen, Z.; Wang, H.; Yeung, D.-Y.; Wong, W.-K.; and Woo, W.-c. 2015. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, 802–810.

Xu, F.; Zhang, P.; and Li, Y. 2016. Context-aware real-time population estimation for metropolis. In *UbiComp*.

Yuan, Q.; Cong, G.; Ma, Z.; Sun, A.; and Thalmann, N. M. 2013. Time-aware point-of-interest recommendation. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, 363–372. ACM.

Zhang, J.; Zheng, Y.; Qi, D.; Li, R.; and Yi, X. 2016. Dnn-based prediction model for spatio-temporal data. In *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 92. ACM.

Zhang, J.; Zheng, Y.; and Qi, D. 2017. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *AAAI*, 1655–1661.

Zonoozi, A.; Kim, J.-j.; Li, X.-L.; and Cong, G. 2018. Periodic-crn: A convolutional recurrent model for crowd density prediction with recurring periodic patterns. In *IJCAI*, 3732–3738.