ST-Attn: Spatial-Temporal Attention Mechanism for Multi-step Citywide Crowd Flow Prediction

Yirong Zhou, Hao Chen, Jun Li*, Ye Wu, Jiangjiang Wu, Luo Chen
College of Electronic Science
National University of Defense Technology
Changsha, Hunan, China
{zhouyirong09, hchen, junli, yewugfkd, jiangjiangwu08, luochen}@nudt.edu.cn

***Abstract*—Multi-step citywide crowd flow prediction (MsCCFP) is to predict the in/out flow of each region in a city in the given multiple consecutive periods. For traffic control and public safety protection, it can provide a long term view for taking measures. However, the spatial and temporal correlations in crowd movements and the lack of information make MsCCFP challenging. In this paper, a deep-learning based prediction model with spatial-temporal attention mechanism is proposed for MsCCFP. The model, called ST-Attn for short, follows the general encoder-decoder framework for modeling sequential data but adopts a multiple-output strategy to preserve the correlations characterizing between each predicted step. The spatial-temporal attention mechanism learns to globally determine the focus on those parts of the city at certain periods that are more relevant to the predicted region and time period. Besides, a pre-predicted result calculated by spatiotemporal kernel density estimation is fed to ST-Attn, which provides a reference for further accurate predicting. Experiments on three real-world datasets are carried out to verify ST-Attn's performance and the results show that ST-Attn outperforms the baselines in terms of MsCCFP.**

***Index Terms*—crowd flow prediction; multi-step ahead prediction; attention mechanism; deep neural network; spatio-temporal analysis;**

I. Introduction

Predicting the movement of crowds in a city is of great importance for traffic control, risk assessment, and public safety protection [1]. Compared with doing next-step prediction, the multi-step citywide crowd flow prediction (MsCCFP) providing long term view is more preferred in practice. As shown in Fig.1, MsCCFP is to predict the in/out flow (the crowds entering/leaving) of each region in a city (divided by rectangular-grids) in the given multiple consecutive periods. Generally speaking, doing earlier and more accurate prediction of citywide crowd flow prevent it from shortsighted or impulsive when taking measures for traffic control and public safety protection [2].

However, the spatial and temporal correlations and the lack of information make MsCCFP challenging. Nowadays, people can conveniently go anywhere in the city due largely to the modern transportation systems. The long-range spatial correlations between regions comes into being in crowd movements and it is difficult to understand the temporal correlations in crowd flow if not viewing many time slices back and forth.

Supported in part by the National Natural Science Foundation of China: 41871284, 61806211.

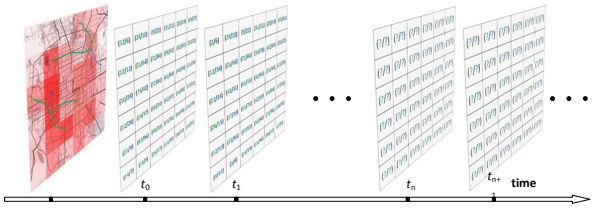


Fig. 1: Multi-step citywide crowd flow prediction.

Besides, for those subsequent steps prediction in MsCCFP, the available information becomes less and less since there is no ground truth from pre-steps.

To tackle such issues, we propose a prediction model based on deep neural networks (DNNs) with spatial-temporal attention mechanism for MsCCFP, called ST-Attn for short. ST-Attn follows the general encoder-decoder framework but with a multiple-output strategy. The contributions of our study are three-fold:

(1) The spatial-temporal attention mechanism (ST-AM) layer. It learns to globally determine the focus on those parts of the city at certain periods that are more relevant to the predicted region and time period. Except for 2-dimensional convolutional neural networks (2D-CNNs) and 1D-CNNs in ST-Attn, ST-AM layers expand the receptive field further to learn the long-range spatial and the long-term temporal correlations in the crowd flow.

(2) Feeding a pre-predicted result to the prediction model. The pre-predicted result is calculated by spatiotemporal kernel density estimation (STKDE)[3] and fed to ST-Attn. It is provided as reference information of future in-out flow filling up the lack of ground truth from pre-steps for those subsequent steps prediction in MsCCFP, which is simple but effective.

(3) Experiments on three real-world datasets are carried out to verify ST-Attn's performance and the results show that ST-Attn outperforms the baselines in terms of MsCCFP.

II. Related Work

Crowd flow prediction can be viewed as a kind of spatial-temporal data prediction problem [2]. Similar work includes taxi passenger demands prediction [2, 4], metro ridership prediction [5, 6], bike-sharing demands prediction [7], geo-sensory time series prediction [8] and so on.

2375-9259/19/$31.00 ©2019 IEEE
DOI 10.1109/ICDMW.2019.00092

Viewing crowd flow as sequential data, many works on MsCCFP are recently inspired by those studies of neuro-linguistic programming [9]. The encoder-decoder frameworks and recurrent neural networks (RNNs) are generally employed due to RNNs' flexibility in dealing with variable length of input and output sequential data. Most existing works use ConvLSTM/ConvGRU, variants of RNNs, as improvement taking both spatial and temporal correlations into consideration [2, 10]. However, the iterative process (each step depends on the previous predicted step) usually makes the training low-efficient and cannot handle long-term correlations well [11].

In the recent researches, attention mechanism shows its success in handling general sequence-to-sequence problems [9]. The intuition behind is that some parts of the input can be more relevant compared to others when generating the output. It expands receptive field directly by constructing long-term/long-range correlations and selecting relevant contents, which conform to the trend of deep learning [11]. Applied on MsCCFP, [2] and [10] introduce attention model to incorporate the representative patterns or periodic patterns of citywide crowd flow into prediction. Although these works are among the first to employ attention mechanism to MsCCFP, the spatial and temporal attention mechanisms are separated or applied only on external features and RNNs are still in use. Different from these methods for MsCCFP, we design hybrid spatial-temporal attention mechanism in the prediction model without RNNs.

III. Problem Definition

As shown in Fig.1, a city is first divided into $p \times q$ grid map according to the longitude and latitude. Thus, a city can be denoted as $\mathbf{G}=[\mathrm{g}_{rc}]_{p\times q}$, where g_{rc} denotes a grid, i.e. a region of the city, lies at the r^{th} row and the c^{th} column.

Definition 1. *Observing time unit. τ is the observing time unit for aggregating the in-out flow count, e.g. 30 minutes. Let $T=[\tau_0, ..., \tau_i, ..., \tau_{n-1}]$ is the whole observing time period.*

Definition 2. *In-out flow. $\boldsymbol{X}_{\tau_i}^{out} = [x_{rc}^{out,\tau_i}]_{p\times q} \in \mathbb{N}^{p\times q}$ records the out-flow count from each grid during time period τ_i. Similarly, $\boldsymbol{X}_{\tau_i}^{in} = [x_{rc}^{in,\tau_i}]_{p\times q}$ records the in-flow count. Let $\boldsymbol{X}_{\tau_i} = [(x_{rc}^{out,\tau_i}, x_{rc}^{in,\tau_i})]_{p\times q} \in \mathbb{N}^{p\times q\times 2}$ stack $\boldsymbol{X}_{\tau_i}^{out}, \boldsymbol{X}_{\tau_i}^{in}$ together as one in-out flow record of time period τ_i.*

***Problem*:** MsCCFP problem. Given the historical observations $\{\mathbf{X}_{\tau_i} | i \in [0, 1, ..., n-1]\}$, predict $\{\hat{\mathbf{X}}_{\tau_j} | j \in [n, n+1, ..., n+l_{out}-1]\}$, aiming to minimize the Root Mean Square Error (RMSE):

$$RMSE = \sqrt{\frac{1}{l_{out}} \sum_{j=n}^{n+l_{out}-1} \| \hat{\mathbf{X}}_{\tau_j} - \mathbf{X}_{\tau_j} \|_2} \qquad (1)$$

where $\mathbf{X}_{\tau_j}$ is the ground truth at τ_j, l_{out} denotes the time steps count to predict.

IV. The Proposed Method

People can conveniently go anywhere in the city due largely to the modern transportation systems. Except for the the regions adjacent to the predicted region and the last observing time unit, the regions further away and the earlier observing time units should receive more or less attention. Due to the uneven activity level in the city and the daily fluctuation, it is the point to determine the focus on those parts of the city at certain periods that are more relevant to the predicted region g_{rc} at the predicted time period τ_j. Inspired by such view, we design the model for MsCCFP in conjunction with spatial-temporal attention mechanism, which is the biggest difference from other MsCCFP models.

Fig.2 shows the architecture of ST-Attn. It predicts the desired l_{out} steps of the in-out flow $\{\hat{\mathbf{X}}_{\tau_j} | j \in [n, ..., n+l_{out}-1]\}$ according to l_{in} last steps $\{\mathbf{X}_{\tau_i} | i \in [n-l_{in}, ..., n-1]\}$. The pre-predicted result calculated by STKDE (see IV-B) $\{\mathbf{P}_{\tau_j} | j \in [n, ..., n+l_{out}-1]\}$ corresponding to the predicted period is used as additional information. The widely used encoder-decoder framework with a multiple-output strategy is adpoted. It first encodes the input sequence of in-out flow tensors into fixed dimensional states and then decodes the states with the pre-predicted result to predict the desired sequence of future in-out flow tensors.

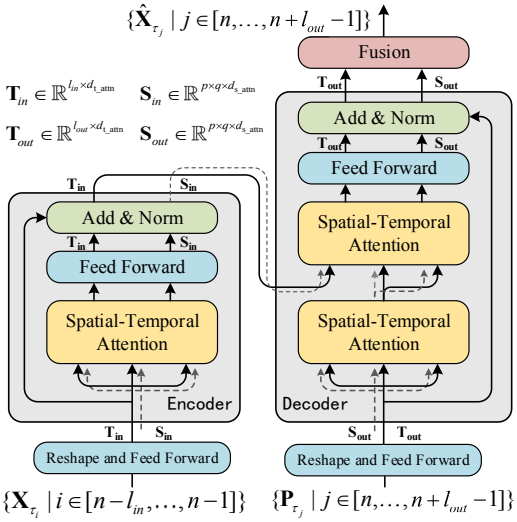


Fig. 2: The architecture of ST-Attn for MsCCFP.

A. Encoder and Decoder

Encoder: The encoder is a stack of a spatial-temporal attention mechanism layer (ST-AM layer, see IV-C and Fig.3) and a fully connected feed-forward layer (FC-FF layer, see IV-D). It outputs two encoded states: $\mathbf{T}_{in} \in \mathbb{R}^{l_{in}\times d_{t\_attn}}$ and $\mathbf{S}_{in} \in \mathbb{R}^{p\times q\times d_{s\_attn}}$, where d_{s_attn} is the specified channels of encoded spatial states and d_{t_attn} is the specified channels of encoded temporal states. $\mathbf{T}_{in}$ maintains all input temporal information with its first dimension equal to l_{in} , while $\mathbf{S}_{in}$ maintains the spatial information with its first two dimensions equal to $p \times q$. A residual connection followed by layer normalization is employed before outputing the encoded states, i.e., LayerNorm(x + ST_FF(x)), where ST_FF(x) denotes the concatenation of the ST-AM layer and the FC-FF layer.

Decoder: Similar to the encoder, it stacks two ST-AM layers and a FC-FF layer. The first ST-AM layer takes the pre-predicted result (see IV-B) as input and outputs two temporary decoded states: $\mathbf{T}_{out}$ and $\mathbf{S}_{out}$. The second ST-AM layer takes the encoded and decoded states as inputs. The implementation of these two ST-AM layers integratedly selects the relevant contents of l_{in} last steps crowd flow information and the pre-predicted result to compute the output. The pre-predicted result is provided as a reference to facilitate the final accurate predicting. A residual connection followed by layer normalization is also employed before outputing the final decoded states: $\mathbf{T}_{out} \in \mathbb{R}^{l_{out} \times d_{t\_attn}}$ and $\mathbf{S}_{out} \in \mathbb{R}^{p \times q \times d_{s\_attn}}$.

To facilitate all the connections, the inputs and outputs of each layer in the encoder are tensors with same shape: $\mathbf{S}_{in} \in \mathbb{R}^{p \times q \times d_{s\_attn}}$ and $\mathbf{T}_{in} \in \mathbb{R}^{l_{in} \times d_{t\_attn}}$. It is similar in the decoder: $\mathbf{S}_{out} \in \mathbb{R}^{p \times q \times d_{s\_attn}}$ and $\mathbf{T}_{out} \in \mathbb{R}^{l_{out} \times d_{t\_attn}}$. Besides, two FC-FF layers with reshape operations are respectively applied on the input sequence of in-out flow tensors and the periodic pattern. Finally, a fusion layer (see IV-E) merges the decoded states: $\mathbf{T}_{out}$ and $\mathbf{S}_{out}$ to output the predicted sequence of future in-out flow tensors.

B. The pre-predicted result calculated by STKDE

For those subsequent steps prediction in MsCCFP, the available information becomes less and less since there is no ground truth from pre-steps. Usually, prediction models based on encoder-decoder framework adopt an iterative process to predict step by step, that is to execute the decoder l_{out} times predicting each step through utilizing the prediction result and hidden state from the previous predicted step, like [2, 12]. However, the errors accumulate as the iterative process going on step by step. Besides, it overlooks the spatial and temporal correlations information in crowd flow from the whole predicted time periods, since only the information of those predicted steps can be utilized [13].

To tackle such issue, we propose to input the decoder with a pre-predicted result, which is calculated by STKDE and provided as reference information of the future in-out crowd flow. Incorporation with the ST-AM layers (see IV-C) and the adopted multiple-output strategy (see IV-E), ST-Attn will have an overall view on the recent, current and future crowd flow while doing MsCCFP.

As a matter of fact, kernel density estimation (KDE) methods are applied to create density surfaces that discribe the spatial distributions of a set of data. To perform KDE on both space and time, we use STKDE proposed in [3] to prepare the pre-predicted result $\{\mathbf{P}_{\tau_j} | j \in [n, ..., n + l_{out} - 1]\}$, where $\mathbf{P}_{\tau_j} = [(p_{rc}^{out,\tau_j}, p_{rc}^{in,\tau_j})]_{p \times q} \in \mathbb{R}^{p \times q \times 2}$. The out-flow estimation result of g_{rc} at τ_j:

$$p_{rc}^{out,\tau_j} = \frac{1}{n h_s^2 h_t} \sum_{i=1}^{n} k_{st}\left(\frac{r - r_i}{h_s}, \frac{c - c_i}{h_s}, \frac{\tau_j - \tau_i}{h_t}\right), \quad (2)$$

where h_s, h_t are kernel bandwidths in space and time, k_{st} is the kernel function [3]. The in-flow estimation p_{rc}^{in,τ_j} is similar without elaboratation due to the limitation of space.

C. Spatial-Temporal Attention Mechanism Layer

An attention mechanism layer can be described as mapping a *query* and a set of *key-value* pairs to an output. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key [11]. That is to say, it selects relevant contents of an input to compute each position of the output representation. Taking each value of the input into consideration to determine the focus, the receptive field of the attention mechanism layer is expanded directly compared with CNNs or RNNs.

Intuitively, as time flies, the out-flow from a region flows into other regions that are farther and farther, and the in-flow probably originates from those regions far away. Due to such dynamic spatial-temporal correlations in the citywide crowd flow, ST-AM layer is designed to learn the correlation between the predicted in-out flow of each grid g_{rc} at τ_j and all inputs in both spatial and temporal domain. It is composed of a temporal attention (T-Attn) branch and a spatial attention (S-Attn) branch (Fig.3).

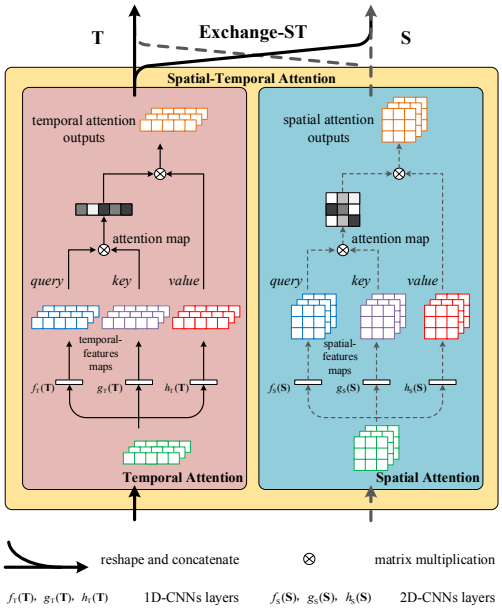


Fig. 3: The Spatial-Temporal Attention Mechanism Layer.

Temporal Attention: The temporal attention branch takes the encoded state $\mathbf{T}_{in}$ or the decoded state $\mathbf{T}_{out}$ as input. They are first transformed into three temporal-feature maps: *query*, *key* and *value* by 1D-CNNs layers (kernel size 3): $f_T(\mathbf{T})$=*1D-CNNs*($\mathbf{T}$), $g_T(\mathbf{T})$=*1D-CNNs*($\mathbf{T}$), $h_T(\mathbf{T})$=*1D-CNNs*($\mathbf{T}$).

The 1D-CNNs layers capture the temporal features locally. Then an attention map is calculated by a softmax layer: Softmax($f_T(\mathbf{T})$,$g_T(\mathbf{T})$), to determine the weight assigned to each value of *query* according to *key*. The temporal attention outputs is Softmax($f_T(\mathbf{T})$,$g_T(\mathbf{T})$)$\cdot h_T(\mathbf{T})$.

Spatial Attention: Similarly, the spatial attention branch takes the encoded state $\mathbf{S}_{in}$ or the decoded state $\mathbf{S}_{out}$ as input.

They are first transformed into spatial-feature maps: *query*, *key* and *value* by 2D-CNNs layers (kernel size 3×3): $f_S(\mathbf{S})$=*2D-CNNs*($\mathbf{S}$), $g_S(\mathbf{S})$=*2D-CNNs*($\mathbf{S}$), $h_S(\mathbf{S})$=*2D-CNNs*($\mathbf{S}$).

The 2D-CNNs layers capture the spatial features locally. Then an attention map is calculated by a softmax layer: Softmax($f_S(\mathbf{S})$,$g_S(\mathbf{S})$), to determine the weight assigned to each value of *query* according to *key*. The spatial attention outputs is Softmax($f_S(\mathbf{S})$,$g_S(\mathbf{S})$)·$h_S(\mathbf{S})$.

Exchange-ST: To mix up information from spatial and temporal domain, the results of S-Attn and T-Attn are exchanged and concatenated. Taking the T-Attn branch in the encoder as an example, state $\mathbf{S}_{in} \in \mathbb{R}^{p\times q\times d_{s\_attn}}$ is first transformed with 2D-CNNs layer (kernel size 1×1) to $\mathbf{S'}_{in} \in \mathbb{R}^{p\times q\times l_{in}}$ and reshaped as $\mathbf{S''}_{in} \in \mathbb{R}^{l_{in}\times pq}$, then concatenated to $\mathbf{T}_{in} \in \mathbb{R}^{l_{in}\times d_{t\_attn}}$. The operations are similar in the S-Attn branch.

D. Fully Connected Feed-Forward

The feed forward layers are used to facilitate the residual connection connections. It consists of a linear transformation with a ReLU activation.

$$FC - FF(\mathbf{x}) = max(0; \mathbf{x}\mathbf{W} + \mathbf{b}) \tag{3}$$

Another way of describing this is as 1D-CNNs (kernel size 1) for the T-Attn branch and 2D-CNNs (kernel size 1×1) for the S-Attn branch.

E. Fusion

Multiple-output strategy: Those models for MsCCFP with RNNs can only predict step by step iterately due to the transmission of hidden states. It overlooks the spatial and temporal correlations information in crowd flow from the whole predicted time periods, since only the information of those predicted steps can be utilized. Since no RNNs in ST-Attn, the multiple-output strategy can be adpoted to predict the future in-out crowd flow of each region at all given consecutive periods at once. That is to preserve the spatial-temporal correlations of crowd flow between regions and between observing time units. Due to the multiple-output strategy adopted, the decoder's input is the pre-predicted result instead of iterately using the pre-steps predicted results. Thus, while doing MsCCFP, the decoder of ST-Attn has an overall view on the recent, current and future crowd flow: i.e., $\{\mathbf{X}_{\tau_i} | i \in [n - l_{in}, ..., n - 1]\}$, [$\mathbf{T}_{in}$, $\mathbf{S}_{in}$] and $\{\mathbf{P}_{\tau_j} | j \in [n, ..., n + l_{out} - 1]\}$ respectively.

Fusion: Finally, to produce $\{\hat{\mathbf{X}}_{\tau_j} | j \in [n, n + 1, ..., n + l_{out} - 1]\}$, the decode states $\mathbf{T}_{out}$ and $\mathbf{S}_{out}$ are directly fused. Concretely, $\mathbf{T}_{out} \in \mathbb{R}^{l_{out}\times d_{t\_attn}}$ is first transformed with 1D-CNNs layer (kernel size 1) to $\mathbf{T'}_{out} \in \mathbb{R}^{l_{out}\times pq}$ and reshaped to $\mathbf{T''}_{out} \in \mathbb{R}^{p\times q\times 2l_{out}}$; $\mathbf{S}_{out} \in \mathbb{R}^{p\times q\times d_{s\_attn}}$ is transformed with 2D-CNNs layer (kernel size 1×1) to $\mathbf{S'}_{out} \in \mathbb{R}^{p\times q\times 2l_{out}}$; then a linear transformation (2D-CNNs with kernel size 1×1) of $\mathbf{S'}_{out}$+$\mathbf{T''}_{out}$ with a ReLU activation and reshape operation is performed:

$$\begin{aligned}&\{\hat{\mathbf{X}}_{\tau_j} | j \in [n, n + 1, ..., n + l_{out} - 1]\}\\ &= Reshape(max[0; 2D - CNNs(\mathbf{S'}_{out} + \mathbf{T''}_{out})])\end{aligned} \tag{4}$$

.

V. Experiments

Experiments to verify ST-Attn's performance on three datasets are conducted in this section.

A. Datasets

The datasets are: (1) Beijing taxi trajectories (**BJTaxi**) [1]: Beijing taxis GPS data; (2) New York City Taxi trip record data (**NYCTaxi**)[14]: the yellow taxi trip records from NYC Taxi and Limousine Commission (TLC); (3) Citibike Trip data from New York City (**Citibike**)[15]: the bike-sharing trip data of Citibike in New York City. The datasets are respectively aggregated as in-out flow according to time interval and grid map size. The details are presented in Table.I.

TABLE I: Details of the datasets.

dataset	Timespan	Time interval	Grid map size
BJTaxi	2013.7.1-2013.10.30 2014.3.1-2014.6.30 2015.3.1-2015.6.30 2015.11.1-2016.4.10	30 minutes	32*32
NYCTaxi	2013.7.1-2016.6.30	1 hour	32*32
CitiBike	2014.1.1-2016.6.30	1 hour	16*16

The last three weeks of each dataset are chosen as testing data, the three weeks before that as validating data, and the rest as training data.

B. Parameter Setting of ST-Attn

- The time steps used to predict l_{in}: 6
- The time steps to predict l_{out}: 6
- The channels of spatial states d_{s_attn}: 16
- The channels of temporal states d_{t_attn}: 64
- The length of weeks used to calculate a pre-predicted result m: 4
- Loss function: Mean Square Error (MSE)
- Optimizer: Adam-optimizer [16]
- Teminated condition: The training reaches 300 iterations, and the model at the epoch that ST-Attn achieves best performance on the validating data is saved as the final prediction model.

C. Baselines & Metric

In order to confirm the effectiveness of ST-Attn, we conduct experiments to compare ST-Unet with five baselines:

- **ST-ResNet**[1]: The first next-step prediction model based on deep residual CNNs for crowd flow prediction. Learned representations are merged in a fusion process along with external information such as date property meta-data and weather data.
- **ST-UNet**[17]: An improved version of ST-ResNet by replacing the deep CNNs structure with UNet structure. It is also a next-step prediction model and better in combination of local-global features.
- **sConvLSTM**[2]: A sequence-to-sequence model for MsCCFP with two layers of ConvLSTM both in the encoder and decoder. There are two layers of 2D-CNNs

following the ConvLSTMs in the decoder, same as ahead of the ConvLSTMs in the encoder.

- **AttConvLSTM**[2]: sConvLSTM with an attention mechanism branch to incorporate the hidden states from the decoder with the crowd flow representative patterns (the clustering result of historical citywide crowd flow reflecting the latent mobility regularities).
- **PCRN**[10]: A prediction model based on pyramidal convolutional recurrent network architecture. It adopts multiple-output strategy for MsCCFP. It uses a loop-back attention mechanism branch to dynamically incorporating the periodic representations of crowd flow, which is the stored hidden states from the top layers of the pyramidal ConvGRUs.
- **ST-UNet+**: ST-UNet with multiple-outupt predicting strategy to do MsCCFP.
- **ST-UNet-**: ST-UNet with direct predicting strategy to do MsCCFP.

To do the multi-step ahead prediction, **ST-ResNet** and **ST-UNet** are tested with iterative predicting strategy. The input and output steps count of **sConvLSTM**, **AttConvLSTM**, **PCRN** are equal to 6, same as **ST-Attn**. The parameter setting of all baselines conforms to those in the corresponding papers. All models are conducted 5 times to measure the their average performance. The metrics we adopt to measure the results is RMSE, as depicted in equation1.

D. Results

Table.II shows the average performance of all prediction models predicting 6-steps output with 6-steps input on the three datasets. In each row, the number colored grey and underlined is the best and second-best results. Table.III shows the parameters count of each model on the datasets. As the grid map of **BJTaxi** and **NYCTaxi** are the same, the prediction models' parameters count on both are the same too (there is a little difference for PCRN).

TABLE II: RMSE of all models performing 6-steps output with 6-steps input.

	BJTaxi	NYCTaxi	CitiBike
ResNet	31.613	8.211	25.805
UNet	30.884	8.101	25.572
ST-Unet+	29.609	7.443	24.74
ST-Unet-	29.806	7.478	25.013
ConvLSTM	27.687	7.657	24.463
AttConvLSTM	27.496	<u>7.412</u>	<u>23.803</u>
PCRN	<u>25.581</u>	9.035	25.814
ST-Attn	25.112	6.598	23.497

In Table.II, it shows that **ST-Attn** outperforms all baselines on the average performance. Certainly, the performance of **PCRN** on **BJTaxi** is almost the same to **ST-Attn**, while **AttConvLSTM** on **NYCTaxi** is almost the same to **ST-Attn**. On dataset **Citibike**, **ST-Attn** performs much better than other baselines, reaching 11% improvement at least beyond the second-best **AttConvLSTM**. Since **PCRN** also performs not so well on dataset **Citibike**, we look into the data. With analysis of **Citibike**, we think that the large relative standard deviation of **Citibike** should be the reason. In Table.III, it shows that the parameters count in **ST-Attn** is much less than that in **AttConvLSTM** and **PCRN**, almost 1/8 of **PCRN**'s parameters count. Compared to **sConvLSTM**, the great amount of parameters in **AttConvLSTM** is due to the attention mechanism branch introducing the crowd flow representative patterns, which includes fully-connected layers. Different from **AttConvLSTM**, **ST-Attn** simply feeds the decoder with a pre-predicted reslut calculated by STKDE. **ST-Attn** is 'slim' but preserves good prediction performance.

TABLE III: The parameters count of each model.

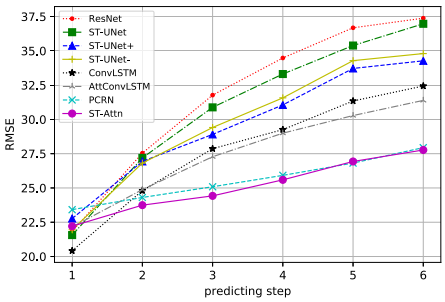
	BJTaxi	NYCTaxi	Citibike
ST-Attn	181376	181376	68480
PCRN	1300912	1300912	957904
sConvLSTM	343130	343130	343130
AttConvLSTM	84859778	84859778	21945218

Fig.4abc show each predicting step's RMSE of all prediction models on the datasets respectively. It shows that **ST-Attn** achieves better performance in most predicting steps. For those first step prediction, ST-Attn is not the best. We think that should be due to the adopted multiple-output strategy. It aims to achieve optimization on the whole instead of on single step. Similar situation can be found in the results of **ST-UNet+** and **ST-UNet-**. The first or second step prediction results of **ST-UNet+** is a little worse than **ST-UNet-**, which adopts a direct predicting strategy to do MsCCFP. For those subsequent steps prediction, **ST-Attn** outperforms other baselines and achieves a better overall results (see Table.II).

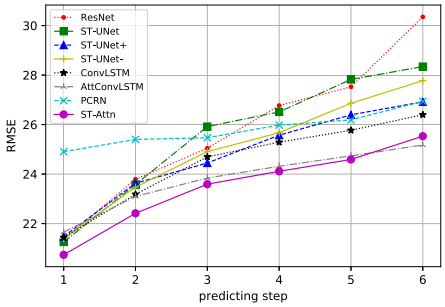
From each predicting step's RMSE, we can figure out that **ST-Attn** outperforms a bit better than **PCRN** due large to the first 3 steps prediction. Similarly, **ST-Attn** outperforms a bit better than **AttConvLSTM** due largely to the first 2 steps prediction. For those subsequent steps prediction, however, the RMSEs tends to be the same. We think it may due to the bad estimation in the later part of the pre-predicted result input to ST-Attn. In the experiments, **ST-Attn** shows its effectiveness and outperforms other baselines though remaining some issues to be further explored.

VI. Conclusion and Discusion

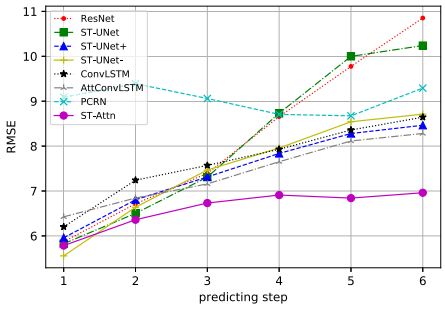
In this paper, we propose a spatial-temporal attention mechanism based model, named ST-Attn, to do the multi-step citywide crowd flow prediction. The general encoder-decoder framework for sequence-to-sequence modeling is adopted. Instead of using RNNs or its variants (ConvLSTM or ConvGRU), a spatial-temporal attention mechanism layer is designed to directly select the relevant contents in both spatial and temporal domain to compute each predicted value, which is equal to gain a global receptive field on the input. Due to no RNNs in our prediction model, a multiple-output predicting



(a)



(b)



(c)

Fig. 4: Each predicting step's RMSE of all prediction models on dataset (a) **BJTaxi**, (b) **NYCTaxi**, (c) **Citibike**.

strategy is adopted and a pre-predited result calculated by STKDE is fed to the decoder to provide a reference for further accurate predicting, which is simple but effective. Compared with several baselines, ST-Attn performs well on the whole in the experiments and shows its effectiveness on MsCCFP. Due to the limitation of space, how the ST-AM layers work and how ST-Attn's variants (such as replacing the decoder's input) would perform differently remains to be explored. In the future work, we will consider how ST-Attn can be modified and applied to do station-level multi-step crowd flow prediction.

References

[1] Junbo Zhang, Yu Zheng, and Dekang Qi. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *AAAI*, pages 1655–1661, 2017.

[2] Zhou Xian, Yanyan Shen, Yanmin Zhu, and Linpeng Huang. Predicting multi-step citywide passenger demands using attention-based neural networks. In *Eleventh Acm International Conference on Web Search & Data Mining*, 2018.

[3] Jay Lee, Shengwen Li, and Shengwen Li. Exploring spatiotemporal clusters based on extended kernel estimation methods. *International Journal of Geographical Information Systems*, 31(6):24, 2017.

[4] Huaxiu Yao, Yiding Liu, Ying Wei, Xianfeng Tang, and Zhenhui Li. Learning from multiple cities: A meta-learning approach for spatial-temporal prediction. *arXiv preprint arXiv:1901.08518*, 2019.

[5] Ziqian Lin, Jie Feng, Yong Li, and Depeng Jin. Deepstn+: Context-aware spatial-temporal neural network for crowd flow prediction in metropolis. AAAI, 2019.

[6] Xiaolei Ma, Jiyu Zhang, Bowen Du, Chuan Ding, and Leilei Sun. Parallel architecture of convolutional bi-directional lstm neural networks for network-wide metro ridership prediction. *IEEE Transactions on Intelligent Transportation Systems*, PP(99):1–11, 2018.

[7] Yi Ai, Zongping Li, Mi Gan, Yunpeng Zhang, Daben Yu, Wei Chen, and Yanni Ju. A deep learning approach on short-term spatiotemporal distribution forecasting of dockless bike-sharing system. *Neural Computing and Applications*, pages 1–13, 2018.

[8] Yuxuan Liang, Songyu Ke, Junbo Zhang, Xiuwen Yi, and Yu Zheng. Geoman: Multi-level attention networks for geo-sensory time series prediction. In *IJCAI*, pages 3428–3434, 2018.

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. 2017.

[10] Ali Zonoozi, Jung Jae Kim, Xiaoli Li, and Gao Cong. Periodic-crn: A convolutional recurrent model for crowd density prediction with recurring periodic patterns.

[11] Sneha Chaudhari, Gungor Polatkan, Rohan Ramanath, and Varun Mithal. An attentive survey of attention models. 2019.

[12] Bowen Du, Hao Peng, Senzhang Wang, Md Zakirul Alam Bhuiyan, Lihong Wang, Qiran Gong, Lin Liu, and Jing Li. Deep irregular convolutional residual lstm for urban traffic passenger flows prediction. *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[13] Lida Mercedes Barba Maggi. *Multi-Step Ahead Forecasting*, pages 49–88. Springer International Publishing, Cham, 2018.

[14] https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.

[15] https://www.citibikenyc.com/system-data/.

[16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[17] Yirong Zhou, Hao Chen, Jun Li, Ye Wu, Jiangjiang Wu, and Luo Chen. Large-scale station-level crowd flow forecast with st-unet. *ISPRS International Journal of Geo-Information*, 8(3):140, 2019.