A Novel Learning Approach for Citywide Crowd Flow Prediction

Xiaoming Yuan[1], Jianchao Han[1], Xue Wang[1], Yejun He[2*], Wenchao Xu[3], Kuan Zhang[4]
[1]Qinhuangdao Branch Campus, Northeastern University, Qinhuangdao 066004, China
[2]College of Electronics and Information Engineering, Shenzhen University, 518060, China
[3]Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Kowloon 999077, Hong Kong
[4]Department of Electrical and Computer Engineering, University of Nebraska-Lincoln, Omaha, NE 68182, USA
E-mails: yuanxiaoming@neuq.edu.cn, {jianchaohan, wangxue}@stumail.neu.edu.cn,
heyejun@126.com*, wenchao.xu@polyu.edu.hk, kuan.zhang@unl.edu

***Abstract*—With the accelerating pace of urban life, population movements are changing dynamically in both space and time. However, the ensuing traffic congestion and potentially dangerous problems caused by crowd flow and increasing uncertain traffic data cannot be ignored. Crowd flow prediction is of critical importance for reducing traffic congestion and eliminating public safety risks in smart city. In this paper, we propose a Spatial Temporal Long-and Short-Term Network (STLSN) model employing deep learning method to predict crowd flow in high precision. The proposed STLSN considers both short and long term temporal dependencies, and it employs local Convolution Neural Network (CNN) to capture spatial correlations between regions. Specifically, we employ the Long Short-Term Memory (LSTM) to capture short term dependency while using recurrent-skip architecture which utilizes the periodic characteristic of flow data to capture the long term temporal dependencies. Moreover, the weighting Point of Interest (POI) is applied to differentiate the importance of function categories. Finally, we conduct experiments on practical dataset and the results demonstrate the effectiveness of our model is over the compared methods.**

***Index Terms*—crowd flow prediction, deep learning, LSTM, smart city.**

I. Introduction

Accurate crowd flow prediction can explore the characteristics of urban residents' activities, which can provide effective decision-making support for the government and management departments, rationally optimizing the allocation of public service resources, providing early warning and dynamic traffic guidance to improve citizens travel experience, and further formulating emergency plans to eliminate public security risks. The crowd flow prediction information can be shared through the short-range communication by efficient vehicular ad hoc network (VANET) communication [1].

Urban crowd flow prediction is a very challenging problem. Firstly, regional population flow data varies with time and space, and thus the prediction model needs to capture dependencies in both time and space dimensions [2]. At the same time, it is also affected by many external factors, such as temperature, events, weather and so on. Wu et al. [3] considered weather, geo-tagged and collision records, which well explains the traffic flow patterns. In recent years, the developing computing and communication technology in big data and artificial intelligence, particularly traditional machine learning and deep learning, provide an opportunity to optimize intelligent traffic systems, reducing traffic congestion and reducing public safety risks [4][5].

Machine learning is used as the traditional method to predict crowd flow. Dong et al. [6] combined wavelets and XGBoost to predict short-term traffic flow, which weakens the high frequency noise of traffic flow. Feng [7] used SVM and multi-kernal computing to model the short term traffic flow adaptively. But they cannot model the traffic flow's complex non-linear correlations enough and their computing capabilities are not enough to train the huge amount of traffic data. Lately, N.G. et al. [8] employed neural network to model non-linear relationships and introduced a simple linear Vector Auto Regressive (VAR) to catch sample spatio-temporal hierarchically. Zhang et al. [9] proposed a ST-ResNet model to predict urban crowd flow by using Convolution Neural Network (CNN) to capture the spatial dependency of each region. DMVST [10] was proposed to capture temporal dependency through using Long Short-Term Memory (LSTM), while the spatial correlations are handled by local CNN.

However, these existing works have their limitations. Firstly, few of studies take region semantic function which is represented by Point of Interest (POI) into consideration. The traveling to a certain place often relies on the functions of the destination [11]. For example, people usually leave home for the office in the morning and go home from office in the evening. Secondly, few of existing studies consider short-term and long-term temporal relations simultaneously. ST-ResNet [9] splits the historical data into three different time periods, but it doesn't explicitly model the complex temporal dependency. DMVST captures the temporal correlations using LSTM, however, it overlooks the long-term temporal dependency.

In this paper, to address above issues, we propose a deep learning model for crowd flow prediction, namely, Spatial-Temporal Long- and Short-Term Network (STLSN). The model's spatial and temporal features are handled by local CNN [10] and LSTM respectively. We regard each categories' importance of POI in a certain region as the weight of corresponding POI distribution and embed them into network, so that to capture semantic features. Additionally, the external influence factors are also taken into consideration. Then the fusion of semantic features, spatial features, and external features is input to recurrent layer to catch the temporal correlations. In the temporal component, recurrent-skip [12] architecture is employed to capture very long-term dependency by the method of making full use of the periodic characteristic of crowd flow data. Our contributions can be summarized as follows:

- We design a semantic component which considers the effect of POI and computes every category's importance of POI in a region as the weight of POI distribution. The designation of self-weighted POI distribution can measure the semantic effect for crowd flow more preciously.
- We propose to use LSTM and recurrent-skip layer in temporal component. This component can capture the short-term and very long-term dependencies jointly.
- We simulate prediction experiments on the practical public dataset with several baselines. Compared with other baselines, the results show that our method has higher accurate.

The rest of this paper is organized as follows. Section II describes the problem definition and some notations. Section III describes the structure of our proposed STLSN model in detail. We discuss and analyze the performance in section IV. Finally, section V concludes the paper.

II. PRELIMINARY

In this section, we briefly introduce the crowd flow prediction and revisit the previous study ST-ResNet[9] to define some notations.

Definition 1 (Region) We regard the whole city as an image and split it into $I \times J$ squares according to latitude and longitude, where each grid has the same size and represents a region. The region can be located by the index (i, j) which represent the i^{th} row and j^{th} column. We define the region set as $R = (r_1, r_2, \ldots, r_{J\times I})$

Definition 2 (Inflow/Outflow) The definitions of inflow and outflow in area (i, j) at t^{th} time interval are as follows :

$$x_t^{i,j,in} = \sum_{Tr\in\mathbb{C}} |\{k > 1 | g_{k-1} \notin (i,j) \wedge g_k \in (\mathrm{i},\mathrm{j})\}| \quad (1)$$

$$x_t^{i,j,\mathrm{out}} = \sum_{Tr\in\mathbb{C}} |\{k >= 1 | g_k \notin (i,j) \wedge g_{k+1} \notin (\mathrm{i},\mathrm{j})\}| \quad (2)$$

where $\mathbb{C}$ represents the set of trajectories at t^{th} time slot. $\mathrm{Tr} : g_1 \to g_2 \to \ldots \to g_{|Tr|}$ is a trajectory in $\mathbb{C}$, $\boldsymbol{g}_k$ represents the geographical coordinates; $g_k \in (i, j)$ means that $\boldsymbol{g}_k$ is located in square (i, j);

For a region $r \in R$ with index of (i, j). The inflow and outflow data at t^{th} time slot of region r can be expressed as $X_t^r \in \mathbb{R}^{I\times J\times 2}$, where $(X_t^r)_0 = x_t^{i,j,in}, (X_t^r)_1 = x_t^{i,j,\mathrm{out}}$.

Problem 1(Crowd Flow Prediction) Given the historical observations $\{x_t^r | t = t-d+1, \ldots, t\}$ until time slot t, we aim to predict the flows X_{t+1}^r at $t+1$ slot. So our problem is formulated as follows: $\boldsymbol{X}_{t+1}^r = \hat{\lambda}\left(\boldsymbol{X}_{t-d+1}^r, \boldsymbol{X}_{t-d+2}^r, \ldots, \boldsymbol{X}_t^r, \boldsymbol{\varepsilon}_{t-d+1}, \boldsymbol{\varepsilon}_{t-d+2}, \ldots, \boldsymbol{\varepsilon}_t\right)$
Where $\boldsymbol{\varepsilon}_{t-d+1}, \boldsymbol{\varepsilon}_{t-d+2}, \cdots, \boldsymbol{\varepsilon}_t$ are the relation factors in each region. $\hat{\lambda}(\cdot)$ is a prediction function.

III. STLSN ARCHITECTURE

In our model, we first fuse the weighted POI and external features with spatial features, then we feed the fusion features into recurrent component to catch the temporal dependencies. Finally, we get the prediction output contains multi influence factors. Figure 1 shows the architecture of our model. It mainly consists of three components, spatial and semantic fusion component, short-term temporal component and long-term temporal component.

A. Spatial and Semantic Fusion Component

There are three elements in spatial and semantic fusion component of STLSN architecture. The specific functions are introduced in the following subsections.

1) Spatial Dependency Component: We use local CNN to take operations on the target region and its surrounding neighborhood, which captures the local spatial dependency. We regard the adjacent area of target region r and r as an $S \times S$ map with two channels including inflow and outflow. We suppose that the data are observed from time $t-d+1$ to t (i.e., $\left\{\left(x_t^{r,in}, x_t^{r,\mathrm{out}}\right) | t = t-d+1, \ldots, t\right\}$). As a result, we get totally $2d$ flow image and concatenate them as $X_t^r \in \mathbb{R}^{S\times S\times 2d}$. Then, the local CNN with K convolutional layers takes X_t^r as input $X_{\mathrm{t}}^{r,0}$, and the computation of convolution at each layer k as follows:

$$X_t^{r,k} = f\left(X_t^{r,k-1} * W_t^k + b_t^k\right) \quad (3)$$

where W_t^k and b_t^k are learned parameters in the k^{th} convolutional layer, $*$ indicates the convolutional operation, and $f(\mathrm{x}) = \max(\mathrm{x}, 0)$ is rectifier activation function.

After K convolution layers, we use a fully connected layer to reduce the dimension of spatial features from $X_t^{r,k} \in \mathbb{R}^{S\times S\times\lambda}$ to a feature vector $\mathrm{S}_t^r \in \mathbb{R}^{S^2\lambda}$ where λ is the number of convolution kernels. So, we get the spatial feature representation $S_{\mathrm{spa}}^r = \left[S_{t-d+1}^r, S_{t-d+2}^r, \ldots, S_t^r\right]$, where $\mathbf{S}_t^r$ is the representation in region r at time interval t.

2) Semantic Features of Weighted POI: The destinations of people's travel always have specific functions, of which types are residences, working places, shopping malls etc represented by POI. In other words, POI as a significant factor affects crowd flow. We also treat the POI distribution is an N-channel matrix $P_{oi} \in \mathbb{R}^{S\times S\times N}$, each channel denotes one type of POI. S is the width of the city map, N is the total categories of POI. In order to differentiate the effect degree of different categories of POI in a region, we need to evaluate the importance of a specific POI in the region units. We utilize the term frequency-inverse document frequency (TF-IDF) which is designed to measure the importance of a word in a document in Natural Language Processing field. In our problem, a target region unit $r \in R$, where R is the collection of all regional units. The TF term of the i^{th} POI type in region

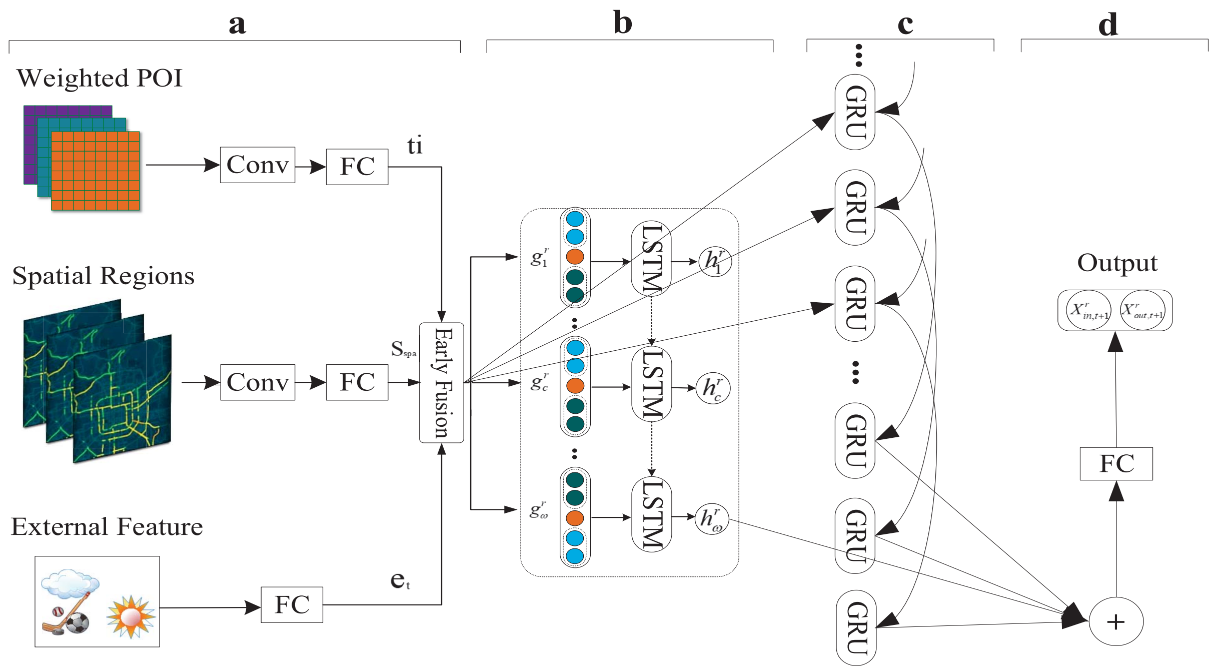


Fig. 1: The Architecture of STLSN. (a)Semantic, spatial and external fusion component. The early fusion of captured features is input to the next temporal component. (b) The short-term temporal capture component is composed of LSTM. (c) The long-term temporal capture component is composed of GRU-based recurrent-skip layer. (d) The final output layer fuses the long-term and short-term dependencies and outputs the predicted flow at t+1 time interval.

r is defined as: $TF_i^r = \frac{p_i^r}{\sum_{i=1}^N p_i^r}$. Where p_i^r denotes the type of i^{th} POI in region r. The IDF terms of the i^{th} POI type in region r is defined as: $IDF_i^r = \log \frac{R}{|r_i|}$. $|\boldsymbol{r}_i|$ is the number of i^{th} category in region r. Then the TF-IDF value of the i^{th} POI type in region r is: $TF\text{-}IDF_i^r = TF_i^r \cdot IDF_i^r$. We use the value of TF-IDF as the weight of POI distribution map. The weighted POI distribution of i^{th} type are computed as: $wP_i = TF\text{-}IDF_i * p_i$. Where wP_i is 1-channel matrix which represents the i^{th} POI type weighted by its corresponding TF-IDF value. $*$ denotes the element-wise product, p_i denotes the i^{th} channel of POI distribution matrix. We concatenate N channels matrix together as the weighted POI distribution: $\mathrm{wP} = \sum_{i=1}^{\mathrm{N}} \mathrm{wP_i}$. Similar to the extraction of spatial features, we also apply local CNN to capture the correlations of target regions POI with its near regions. And through fully connected layer to reduce the dimension. Finally, we get the weighted POI features ti.

3) External features: We stack fully connected layers to embed external features and increase the dimension. The output of external component denotes as e_t^r. Finally, we early fuse the spatial features, semantic features, and external features before input to the next temporal component. The fusion result denotes as:

$$g_t^r = S_{spa}^r \oplus ti \oplus e_t^r \tag{4}$$

B. Short and Long Term Temporal Component

The features captured from above components are not static. In order to catch the temporal dependency, we use LSTM network as our choice. We consider ω time slots in a day, the output feature h_t^r at t time slot is processed as follows:

$$h_{s,t}^r = LSTM\left(g_t^r, h_{t-1}^r\right) \tag{5}$$

Then we get the short-term temporal information $h_{s,t}^r = h_t^r$. However, above LSTM component only takes short time intervals into consideration. LSTM usually hard to catch long term dependency due to gradient vanishing. If we predict the crowd flow at t time slot (e.g. 9:00am to 9:30am) for a day, besides most recent records, the leverage of historical days records at the same time slot is also important. In order to solve this issue, we use recurrent-skip layer which extends temporal span of information pass and eases the optimization process through adjacent period links. The updating process described as follows:

$$h_t^r = GRU\left(g_t^r, h_{t-p}^r\right) \tag{6}$$

where the input of this layer is g_t^r, p is the skip length or the period length of the dataset.

The d hidden states express as $h_{t-d+1}^r, h_{t-d+2}^r, \ldots, h_t^r$ due to the flow happens from time $t-d+1$ to t. Then, we get the long-term periodic information through dense layer which is expressed as follows:

$$h_{l,t}^{r} = \sum_{i=0}^{t-d-1} W_i^r h_{i-p}^{r} + b \tag{7}$$

Finally, we concatenate the short-term information $h_{s,t}^{r}$, long-term information $h_{l,t}^{r}$:

$$h_{o,t}^{r} = h_{s,t}^{r} \oplus h_{l,t}^{r} \tag{8}$$

Then we feed $h_{o,t}^{i}$ to a fully connected layer to get the final prediction of inflow and outflow for region r at time interval $t+1$, which is denoted as $X_{in,t}^{i}$ and $X_{o,t}^{i}$ respectively. The final operation is defined as:

$$X_{t+1}^{r} = \left[X_{in,t+1}^{r}, X_{o,t+1}^{r}\right] = \tanh\left(W_o^r h_{o,t}^{r} + b_{o,t}^{r}\right) \tag{9}$$

where W_o^r and $b_{o,t}^{r}$ are learned parameters.

C. Objective Function of STLSN Architecture

$$\ell(\theta) = \sum_{\xi=1}^{n} \eta \left(X_{t+1}^{\xi} - \hat{X}_{t+1}^{\xi}\right)^2 \tag{10}$$

where θ is a collection of all parameters in this network, ξ is the total number of data samples. We use TensorFlow and Keras to implement our model and choose Adam to optimize the training process.

D. The Training Process of STLSN Architecture

The training process of STLSN is shown in Algorithm 1. We build training examples based on the original sequence of the crowd flow maps and POI distributions (lines 1-11), including S_{spa}, $h_{s,t}^{r}$, $h_{l,t}^{r}$, represent the local spatial features, short-term temporal features, long-term temporal features respectively.

Algorithm 1 STLSN Training Procedure

Input: Historical data: $\{\mathcal{X}_0^r \dots \mathcal{X}_n^r\}$; Time interval vector: $\{t-d+1, \dots, t\}$; External features: $e_{t-d+1,\dots,t}^{r}$; POI distribution: P_{oi}.
Output: Learned STLSN model

1: $\Omega \leftarrow \varnothing$
2: compute the weighted POI distribution ti.
3: **for** $\forall r \in R$ **do**
4: **for** all available time slot $\{t-d+1, \dots, t\}$ **do**
5: $S_{\text{spa}} = \left[X_{t-d+1}^{r}, X_{t-d+2}^{r}, \dots, X_{t}^{r}\right]$
6: $h_s = \left[h_{s,t-d+1}^{r}, h_{s,t-d+2}^{r}, \dots, h_{s,t}^{r}\right]$
7: $h_l = \left[h_{l,t-d+1}^{r}, h_{l,t-d+2}^{r}, \dots, h_{l,t}^{r}\right]$
8: put training instance
9: $\left(\left\{S_{\text{spa}}, h_{s,t}^{r}, h_{l,t}^{r}, ti, e_t^r\right\}, X_{t+1}^{r}\right)$ into Ω;
10: **end for**
11: **end for**
12: initialize all learnable parameters θ in STLSN;
13: **repeat**
14: select a batch of instances Ω_i from Ω randomly
15: optimize θ by minimizing objective function
16: using Adam and Ω;
17: **until** meet the stop criteria;

IV. Experiments

In this section, we introduce dataset and contrast baselines. In addition, we outline the evaluation metrics and parameter settings of the model. Finally, we discuss the performance of the model.

A. Dataset

We use the public practical dataset BikeNYC which contains trajectories and weather. The collection of POI types for this dataset is 9. They are Food, Residence, ShopServic, College, NightlifeSpot, TravelTransport, ArtEntertainment, ProfessionalOtherPlace, OutdoorsRecreation. This dataset is the Bike rent information that happened in 2014, from Apr.1st to Sept. 30th in New York City. The data of the past 14 days was used as testing data and the remaining data as training data.

B. Baselines

We compare our STLSN model with the following baselines:

HA: It predicts flows of crowd by the average value of historical data.

ARIMA: Auto-Regressive Integrated Moving Average (ARIMA) is a classical method that combines moving average and autoregressive components for modeling time series.

VAR: Vector Auto-Regressive (VAR) is a traditional temporal modeling method which can capture the multiple relationships among several flow time series.

ConvLSTM [13]: It is a combination of CNN and LSTM, who takes spatial and temporal influences into consideration.

ST-ResNet [9]: It is a novel spatial-temporal neural network based on CNN only.

C. Metrics

We use two metrics as the evaluation of our model, they are Rooted Mean Square Error (RMSE) and Mean Average Percentage Error (MAPE) which are defined as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\hat{X}_{t+1}^{r} - X_{t+1}^{r}\right)^2} \tag{11}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{\left|\hat{X}_{t+1}^{r} - X_{t+1}^{r}\right|}{X_{t+1}^{r}} \tag{12}$$

where n is the quantity of all samples, $\hat{\boldsymbol{X}}_{t+1}^{i}$, $\boldsymbol{X}_{t+1}^{i}$ represent the real value and prediction value of region r at time slot $t+1$ respectively.

TABLE I: Performance and Comparison of different methods

Method	RMSE	MAPE
Historical average	13.10	28.04%
ARIMA	11.74	26.34%
VAR	12.16	27.41%
XGBoost	9.83	23.25%
ConvLSTM	10.27	24.14%
ST-ResNet	9.67	24.06%
STLSN	9.43	21.67%

D. Parameters and Settings

The span of each time slot is 30 minutes. We normalized the inflow and outflow values for all regions to [0,1] by using Min-Max normalization method. As the output values of our model in the range of [0,1], we denormalize the values and use it for evaluation. Due to the discrete of external features, one-hot encoding is applied to convert into vector.

We split 80% of the dataset as a training set, and the remaining 20% as validation set. For near spatial component, the size of each neighborhood considered is set as 7×7 (i.e., S=7). Additionally, we set $K = 3$ (the number of convolution layers), convolution kernel size to 3×3, $\lambda = 64$ (number of kernels), $\alpha = 64$ (the dimension of output). For temporal component, we set $\kappa = 7$ (i.e., previous 3.5 hours) for short-term LSTM, $\sigma = 3$ (i.e., previous 3 days) for long-term GRU, $p = 24$ (i.e., the skip length is 24 hours). The dimension of hidden output of LSTM is 128. The batch size in training process is set as 64.

E. Performance and Comparison

Table I shows the performances of our model and baselines. Our proposed model STLSN has the lowest values of RMSE and MAPE compared with above baselines. The traditional time series prediction methods such as HA, ARIMA perform not well. Because they only depend on the historical data of prediction value and ignore spatial and other semantic features. VAR also ignores the relevant features except the predicting value itself, which leads to bad performance. The XGBoost achieves better performance compared with traditional methods, due to its optimization train process. For deep learning methods, ConvLSTM integrates convolutional operation with LSTM units to capture both spatial and temporal information. ST-ResNet uses CNN to capture spatial information in closeness, period and trend, but overlook the temporal sequential dependency. Our model not only considers the nonlinear temporal dependencies and spatial features but also semantic and external information, so that it outperforms these baselines.

F. Effectiveness of semantic and long-term temporal component

We define the variants of STLSN model to study the effectiveness of each component.

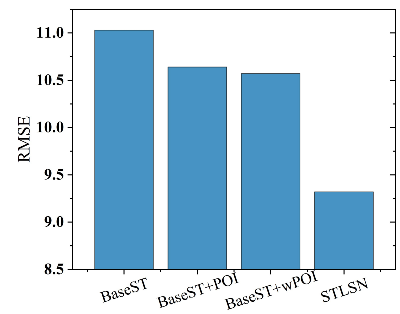
BaseST: this variant consists of spatial component and short-term temporal component only.

BaseST + POI: In this variant, we concatenate the POI distribution which is only the distribution indicator without weight and spatial representation. Then we feed it into LSTM as semantic spatial features to catch temporal dependency.

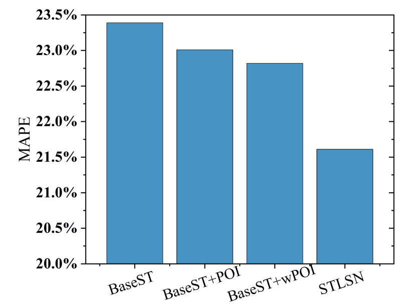
BaseST + wPOI(weighted POI): In this variant, POI distribution is mapped into weights by TF-IDF regular. We concatenate the weighted POI distribution with spatial representation. Then we feed it into LSTM.

BaseST + LT(long-term temporal component): In this variant, the spatial representation is put into LSTM and Recurrent-skip layer to catch short-term temporal dependency and long-term dependency respectively.

STLSN: Our proposed model, which consist of spatial, semantic, external, long and short term components.



(a)



(b)

Fig. 2: (a) RMSE with respect to the effect of semantic component. (b) MAPE with respect to the effect of semantic component.

Figure 2 and 3 show the performances of STLSN and its variants. BaseST + POI and BaseST + weighted POI outperform BaseST, because BaseST overlooks the semantic influence of human mobility. Additionally, Base + weighted POI has lower RMSE and MAPE, because it computes the proportion of each type of POI in a region. The comparison result proves the effectiveness of considering the importance of each POI in a region. The BaseST + LT performs better than BaseST, because BaseST overlooks the long term dependency. The proposed model STLSN outperforms all its variants, which demonstrates the effectiveness of considering semantic and long term temporal correlation factors.

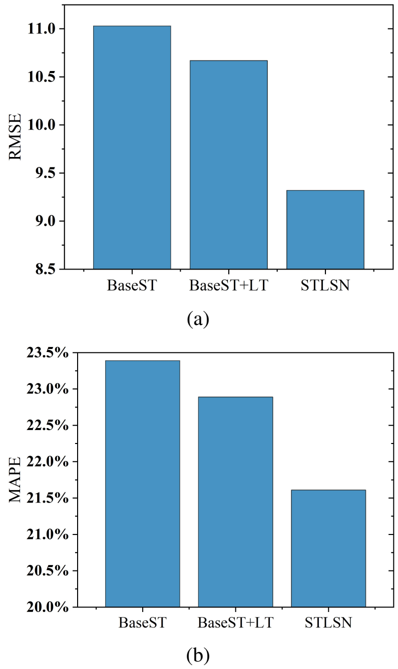


Fig. 3: (a) RMSE with respect to the effect of long term component. (b) MAPE with respect to the effect of long term component.

V. Conclusion and Discussion

We have proposed an STLSN model for crowd flow prediction in a city, which achieves long term temporal dependency and takes the effect of semantic attributes into consideration, and combined them properly. The STLSN model is mainly composed of local CNN, LSTM, and GRU. We have conducted our experiment on a large-scale practical dataset. The experiment results have shown that the performance of STLSN model is significantly better than several baselines, which confirms the effectiveness of our model. In the future, we will focus on the question that how to predict the flow in a fine-grained region with higher accuracy.

Acknowledgment

This work is supported in part by the National Natural Science Foundation of China under Grants 61372077, 61801299, 61901099 and 61871433, in part by the Shenzhen Science and Technology Programs under Grants GJHZ20180418190529516, JCYJ20170302150411789, JCYJ20170302142515949, and GCZX2017040715180580, and in part by the Guangzhou Science and Technology Program under Grant 201707010490 and in part by the Fundamental Research Funds for the Central Universities (N182303032).

References

[1] Tang Y., Cheng N., Wu W., et al. Delay-Minimization Routing for Heterogeneous VANETs With Machine Learning Based Mobility Prediction. *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3967-3979, Apr. 2019.

[2] Zhang J., Zheng Y., Qi D., et al. DNN-based Prediction Model for Spatio-temporal Data. in *Proc. The 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, Burlingame, California, Oct. 31 - Nov. 03, 2016, pp. 92-95.

[3] Wu F., Wang H., Li Z. Interpreting Traffic Dynamics Using Ubiquitous Urban Data. in *Proc. The 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, Burlingame, California, Oct. 31 - Nov. 03, 2016, pp. 69-72.

[4] Li J., Fu D., Yuan Q., et al. A Traffic Prediction Enabled Double Rewarded Value Iteration Network for Route Planning. *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 4170-4181, Jan. 2019.

[5] Lei L., Chen C., Qiu T., et al. A Data Dissemination Scheme Based on Clustering and Probabilistic Broadcasting in VANETs. *Veh. Commun.*, vol. 13, pp. 78-88, Jul. 2018.

[6] Dong X., Lei T., Jin S., et al. Short-Term Traffic Flow Prediction Based on XGBoost. in *Proc. IEEE 7th Data Driven Control and Learning Systems Conference (DDCLS)*, Enshi, China, May 2018, pp. 854-859.

[7] Feng X., Ling X., Zheng H., et al. Adaptive Multi-Kernel SVM With SpatialCTemporal Correlation for Short-Term Traffic Flow Prediction. *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 6, pp. 2001-2013, Sep. 2018.

[8] Polson N. G, Sokolov V. O. Deep Learning for Short-term Traffic Flow prediction. *Transp. Res. Part C: Emerging Technol.*, vol. 79, pp. 1-17, Jun. 2017.

[9] Zhang J., Zheng Y., Qi D. Deep Spatio-temporal Residual Networks for Citywide Crowd Flows Prediction. in *Proc. Thirty-First AAAI Conference on Artificial Intelligence*, San Francisco, California, USA , Feb. 2017. pp. 1655-1661.

[10] Yao H., Wu F., Ke J., et al. Deep Multi-view Spatial-Temporal Network for Taxi Demand Prediction. in *Proc. Thirty-Second AAAI Conference on Artificial Intelligence*, New Orleans, Lousiana, USA, Feb. 2018, pp. 2588-2595.

[11] Xu F., Zhang P., et al. Context-aware Real-time Population Estimation for Metropolis. in *Proc. The 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, Heidelberg, Germany, Sep. 2016, pp. 1064-1075.

[12] Lai G., Chang W. C., Yang Y., et al. Modeling Long- and Short-term Temporal Patterns with Deep Neural Networks. in *Proc. The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*, Santiago, Chile, Aug. 2018, pp. 95-104.

[13] Xingjian S. H. I., Chen Z., Wang H., et al. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. in *Proc. Advances in Neural Information Processing Systems Conference*, Montreal, Canada, Dec. 2015, pp. 802-810.