ST-DRN: Deep Residual Networks for Spatio-Temporal Metro Stations Crowd Flows Forecast

Yang Ning*, Yang Huang*, Jinyang Li*, Qi Liu*, Disheng Yang*, Wei Zheng† and Hengchang Liu†
University of Science and Technology of China
The Comprehend Company
Suzhou, Jiangsu Province, China
{ning1992, sa615313, ljyustc, liuqi100, jkfdre}@mail.ustc.edu.cn*
wei.zheng@comprehend.com.cn†
hcliu@ustc.edu.cn†

***Abstract*—Forecasting the inflow and outflow of crowds at metro station, immediately controlling the number of people entering at some special times and places to avoid the occurrence of malignant events for public safety, is of great significance to subway stations management and very challenging as it is affected by many complex elements, such as inter region station flows, major events or activities, and weather. We propose a approach based on deep residual learning, called ST-DRN, to discern the pattern of spatial and temporal and integrally predict the inflow and outflow of crowds in each subway station of a city. We propose an end-to-end structure of ST-DRN based on distinct attributes of spatio-temporal data. More specifically, we apply the residual neural network framework to model the temporal nearby, day, and week properties of crowd in subway station. For each feature, we design a branch of residual convolutional units, each of which handles the spatial properties of subway crowd. ST-DRN learns to dynamically summation the output of the three residual neural networks, assigning different weights to each branch. The summation is also further combined with external elements, such as weather, holiday and workdays or weekends, to forecast the final traffic flow of crowds in each station. Evaluations on the automatic fare collection (AFC) system subway record data in Suzhou demonstrate that we proposed ST-DRN outperforms than three prominent baseline methods.**

***Index Terms*—public safety, deep residual learning, flow forecasting, ST-DRN**

I. Introduction

Predicting crowd flow in a city or some focused area has play an importance role in traffic management and public safety. With sudden increase of urban population and the rapid development of urban public transportation, urban passenger flow quantity increases sharply [1]. The urban subway transportation has brought huge passenger management pressure to the surrounding stations, especially when the sudden crowded passenger flow gathered and dissipated in a one-time special event. For one case, massive crowds of people streamed into the Bund, landing famous attractions and financial center of Shanghai, at the 2015 for celebrating New Years Eve, resulting in a catastrophic stampede that killed 36 people. If we can predict the flow of crowd in subway station, such tragedies can be mitigated or prevented by utilizing emergency mechanisms, such as conducting traffic flows control, sending out warnings, or evacuating people, in advance.

In this paper, we forecast two styles of crowd flow in a subway station: inflow and outflow, using the records collected by automated fare collection (AFC) subway system [2] in Suzhou, China. However, predicting the flow of crowd in each station of subway system at a city simultaneously is very challenging, influenced by the following three complex elements:

- **Spatial elements:** The outflow of a station is affected by inflow of nearby stations as well as distant stations. But the inflow of a station associate with the passenger travel habits, changing with temporal factors and external factors mentioned below.
- **Temporal elements:** The flow of crowds in a subway station is affected by recent time intervals, both near and far. For example, the flow overcrowding occurring the inflow at 8:00am in someone station will affect the outflow that of 9:00am in other stations, typically for the worker taking subway to work from residential area to work area and the opposite situation occurring beginning at 5:00pm. In addition, the commutes during rush hours may be similar on consecutive workdays, repeating every day. In addition, morning rush hours may gradually happen later with the winter coming. As the temperature gradually decreases and the sun rises later in the day, people get up later and later.
- **External elements:** Some external factors, such as weather conditions and holiday may influence the flow of crowds extremely in some special station. In related

part, we show the external factors figure, using one month data to draw, to explain how it effect the flow at work area and resident area nearby station.

To deal with these factors, we employ a spatio-temporal deep residual network (ST-DRN) to collectively predict inflow and outflow of crowds in metro stations. Our main dedications are at four sections:

- ST-DRN employs convolution-based residual networks to model nearby and distant spatial factors between any two stations within a certain time frame, while ensuring the models forecast accuracy is not consisted by the deep structure of the neural network.
- We summarize the temporal characteristics of crowd flows into three classes, consisting of temporal nearby, day and week. ST-DRN respectively employs three residual networks to model these characteristics.
- ST-DRN aggregates the output of the three beforementioned networks, dynamically, assigning different weights to different branches and parts. The aggregation is further combined with external factors (e.g., holiday).
- We evaluate our approach using Suzhou AFC subway system record data. The results demonstrate the advantages of our approach compared with some baselines. In particular, the forecasting precision of L4 is promoted than that of ARIMA [3] by 55.94%. Furthermore, considering the data collected not be timely, we propose using data of earlier one hour than predicted time t and only adjust the parameters of nearby, day and week to indicate our model practical value on real subway system.

The remainder of this paper is organized as follows. We discuss the related past work in Section II followed by a complete description of our ST-DRN model in Section III. Finally, we demonstrate our experimental results in Section IV and give a brief introduction of conclusion and future work in Section V.

II. Related Work

In this section, we briefly introduce several existed methods on predicting flows and slightly introduce deep residual learning.

A. Previous Work

There are some previously published works on predicting flows in metro based on historical passenger data collected in AFC system to improve on advertising efficiency at subway system [4] and also to reduce last-mile distances from metro station to destination and decrease travel time using cellphones, vehicles and smartcard data [5]. They predict millions or billions of individuals mobility traces rather than the aggregated crowd flows in an area. Some other researchers predict travel traffic flow on the road by joint probability distribution between the cause nodes (data utilized for forecasting) and the effect node (data to be forecasted) in a constructed Bayesian network [6]. Most of them are predicting single or multiple road segments, rather than citywide ones. These work are not considering the dependencies among areas and using history data not effectively and comprehensively than us.

CNNs have been universally applied to various problems, especially displaying unparalleled effect in the field of computer vision [7]. Residual learning allows such networks to have a very super deep structure [8] and recurrent neural networks (RNNs) [9] have been generally employed for sequence learning tasks [10]. However, both kinds of neural networks can only capture spatial or temporal dependencies.

B. Deep Residual Learning

Deep residual learning [8] allows convolution neural networks to have ultra-deep structures in excess of 100 layers, even over 1000 layers, still displaying compelling accuracy and convinced convergence behaviors. Furthermore this method has shown compelling results on diverse challenging identification tasks, including image classification, object detection, segmentation and localization [8].

In formal, a residual unit [11] with an identity mapping is defined as:

$$X^{(l+1)} = X^{(l)} + F\left(X^{(l)}\right) \tag{1}$$

where $X^{(l)}$ and $X^{(l+1)}$ are the input and output of the lth residual unit, respectively; F is a residual function, a stack of two 3×3 convolution layers [11]. The core idea of the residual learning is to learn the additive residual function F with respect to $X^{(l)}$ [11].

III. ST-DRN Model

Fig. 1 display the structure of ST-DRN model, which is consisted of four main parts modeling *nearby*, *day*, *week* and *external* elements, respectively. As demonstrated in Fig. 1, first we translate Inflow and outflow throughout all city metro subway system station at each time interval into a 2-channel array-like tensor, using the method introduced in below. Then we divide the time axis into three segments before the predicted time, representing *nearby* effects, *day* history and *week* history. Next the 2-channel flow tensors of slots in each time segment are fed into the first three parts, sharing the same network structure with a one-dimensional convolutional neural network allowed by a Residual Unit sequence, respectively to model the before mentioned three temporal factors: *nearby*, *day* and *week*, separately. This structure captures the spatial dependency between nearby and distant stations by one-dimensional convolution. In the external component, we manually collect some features from external data sets, such as holidays, weather conditions and workdays or weekends, feeding them into a two-layer fully-connected neural network. The results of the first three parts are fused as X_{RU} based on parameter matrices, which assign different weights to the outputs of different components in different parts. X_{RU} is further combined with the output of the external component X_{Ex}. In the end, the integration is mapped into [-1, 1] by a Tanh function, which yields a faster convergence than the standard logistic function in the process of back-propagation learning [12]. In the next some subsections, we give the

evidence to prove the spatial, temporal and external influence to the station crowed flows.

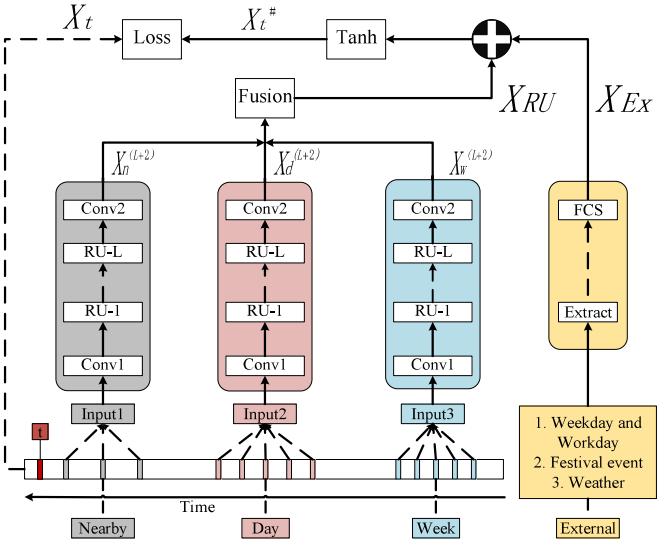

Fig. 1. The structure of ST-DRN. Conv:one-dimensional convolution; RU:residual unit; FC:fully-connected

A. Crowed Flows in Metro Station

In this study, we predict inflow and outflow of all metro stations in a city, as shown in Fig. 3, based on a sequence in all subway lines stations placed into an array K one by one. We partition a record in AFC system, the format of it in Table I, into an inflow and outflow depended on the in information (i.e. In_date, In_station and In_time) and out information (i.e. Out_date, Out_station and Out_time). As in before 6:00 and after 23:00, closing the time to start and stop of subway train in city, most of stations have no crowd flows, in the paper, we predict the flow from 6:00 to 23:00 and use 30 minutes as a time slot. And its also possible for no flows in a station that we fill zero as the flows number for training or testing ST-DRN model. After preprocessing, we show the number of statistics using half hour as a slot in one station at January 2017 in Table II.

TABLE I
DATA FORMAT IN SUZHOU METRO CARD RECORD

Card_no	In_date	In_station	In_time
1	2	3	4
Out_date	Out_station	Out_time	
5	6	7	

The inflow and outflow in all K stations can be represented as a tensor $X_t \in R^{K\times 2}$ where $(X_t)_{k,0} = x_t^{k,in}$, $(X_t)_{k,1} = x_t^{k,out}$ at the tth time slot. For this dynamic system over a spatial station point denoted by array K, there are inflow and outflow at each station point over each time slot. Thus the data mined by us to be a format $X \in R^{K\times 2}$ tensor at any time. And given the historical data $\{X_t | t = 0, ..., i-1\}$, in this paper, we forecast X_t by using different time slot data.

B. Residual Unit

As we all well known, large deep convolutional networks compromise training effectiveness though the outstanding activation function (e.g. ReLU) and regularization techniques are utilized [7], [13]. And we still need a deep network to tackle very large metro stations dependencies, specially if the number of station is large in a city. For metro stations' flows data of a city, typically, presume that the number of stations is 59 , and the kernel size of one-dimensional convolution is used to 40, different size with different performance of model depending on the length of most people taking subway time and stations' layout in a city as we use one-dimensional convolution. If we want to capture all metro stations' dependencies (i.e., each node in high-level layer depends on all nodes of the input), it needs more than 10 succeeding convolutional layers. To handle this problem, we apply residual learning [8] in model, showing very effective for training super deep neural networks of more than 1000 layers.

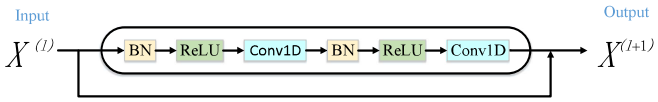

Fig. 2. Residual Unit; BN: batch normalization; ReLU: activation function; Conv1D: one-dimensional convolution

Residual Unit Function and equation:

$$X^{(l+1)} = X^{(l)} + F\left(X^{(l)}; \theta^{(l)}\right), l = 1, \cdots, L \qquad (2)$$

where F is the residual function (i.e. consists of ReLU + Convolution, see F in Fig. 2), and $\theta^{(l)}$ includes all learnable parameters in the lth residual unit. We also employ Batch Normalization (BN) [13] that is placed before ReLU, as shown in Fig. 2.

C. Spatio Elements

A city usually has a very large size, containing many metro stations with different distances. Intuitively, the inflow of crowds in a station may affect other stations' outflow depending on the most of people taking ride time length, which can be effectively processed by the convolutional neural network (CNN) that has shown its powerful ability to hierarchically handle the spatial structural information [12]. Furthermore, subway systems connect two stations with a very far distance, leading to the dependency between distant stations. In order to process the spatial dependency of any station, we need to construct a CNN with many layers because one convolution only handling spatial near dependencies, delimited by the size of their kernels. The same issue also has been found in the video sequence generating task where the input and output have the same resolution [14]. Several methods have been introduced to avoid the loss of resolution brought about by sub-sampling while preserving distant dependencies [10]. Different from the traditional CNN, we do not apply the sub-sampling (i.e. pooling), but only convolutions [15]. We discover that a node in the high-level feature map depends on

some nodes, relying on the size of convolution kernel, of the middle-level feature map, those of which depend on all nodes in the lower-level feature map (i.e. input). It means that one convolution naturally handles station near dependencies, and deep layers convolutions can further handle distant even all stations' dependencies.

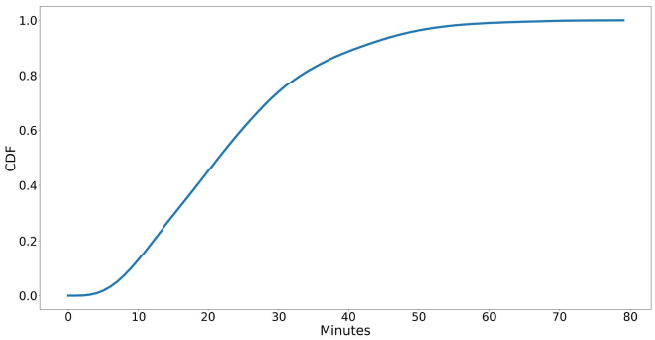

Fig. 3. CDF of taking ride time length in SuZhou (January 2017).

In this paragraph, we discuss how to chose the kernel size of one-dimensional convolution. We draw the figure of metro stations distribution at Suzhou, Showing in Fig. 4, and mark the number for each station, the cross station number the N-9 in line1 and number N-38 in line2. According to Fig. 3, we can summary that the most taking ride time length range from 10 minutes to 40 minutes at Suzhou. Removing the waiting time and transferring time, the maximum number of passengers through the station number is about 18. Obviously, The direction for the max value of station number is boarding in N-0 and leaving in N-42, actually just pass 14 stations in spatial. Considering the waiting time and transferring time, the average subway departure time length, we chose about 40 as the best kernel size. And we will give different length experiment results in evaluation part to prove our discussion.

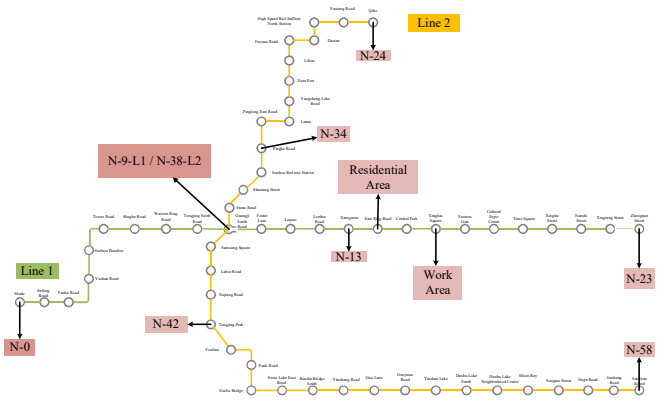

Fig. 4. The stations' distribution of Suzhou subway system.

D. Temporal Elements

As shown in Fig. 1, the first three parts (i.e. *nearby*, *day*, *week*) share the same network structure, which is consisted of two sub-components: one-dimensional convolution and residual unit. Fig. 5 (a) and (d) show the ratio curves using Suzhou subway data presented in Table II where x-axis is time gap between two time slots and y-axis is the average ratio value between arbitrary two inflows at the same time gap. The curves from two different areas, marked in Fig. 4, all show an empirical temporal conjunction in time series, in other words, inflows of close time slots are more relevant than ones of distant time slots, which implies temporal nearby. The two curves have different shapes, which demonstrates that different areas ave different features of nearby, particularly in morning and evening work peak. Fig. 5 (b) and (e) show flows at all time slots of 7 days. Distinctly we can see the daily periodicity in both areas, because of having trend fluctuation of week. In work area, the peak values on workdays are much higher than ones on weekends. Resident area has similar peak values for both workdays and weekends. Fig. 5 (c) and (f) describe sum of flows of one week at January 2017. As time goes by, the flows sharply decreases in work area and resident area, as the last week with Spring Festival, vary important for Chinese and many people moving from city serval days in advance. It shows the different trends in different areas. In conclusion, flows of this two areas are all influenced by *nearby*, *day*, and *week*, but the extents of affect may be very different. This situation is also applicable to other stations.

The *nearby* part in Fig. 1 employs a few 2-channel flows tensor of in slots prior the forecasting time, we want, to pattern temporal nearby dependence. Let the nearby segment be $[X_{t-l_n},\ X_{t-(l_n-1)},\ \ldots,\ X_{t-1}]$, called as the closeness dependent sequence. We first bind them along with the first axis (i.e. time interval) as one tensor $X_n^{(0)} \in R^{K\times 2l_n}$, which is followed by a one-dimensional convolution (i.e. Conv1 shown in Fig. 3) as:

$$X_n^{(1)} = f\left(W_n^{(1)} * X_n^{(0)} + b_n^{(1)}\right) \tag{3}$$

where * represents the convolution1; f is an activation function, e.g. the rectifier $f(x) := max(0, x)$ [7]; $W_n^{(1)}$, $b_n^{(1)}$ are the learnable parameters in the first layer. In our ST-DRN (see Fig. 3), we adopt L residual units on Conv1 as follows. On top of the L^{th}residual unit, we append a one-dimensional convolutional layer (i.e. Conv2 shown in Fig. 1). With 2 convolutions and L residual units, the output of the nearby part of Fig. 1 is $X_n^{(L+2)}$.

Moreover, As the same procedures mentioned above, we build the *day* and *week* parts in Fig. 1. Assume that there are l_d time slots from the day segment and the week is w. Therefore, the day dependent sequence is $\left[X_{t-l_d.d}, X_{t-(l_d-1).d}, \cdots, X_{t-d}\right]$. With the convolutional operation and L residual units like in (2) and (3), the output of the day part is $X_d^{(L+2)}$. Analogously, the output of the week part is $X_w^{(L+2)}$ with the input $\left[X_{t-l_w.w}, X_{t-(l_w-1).w}, \cdots, X_{t-w}\right]$ where l_w is the length of the week dependent sequence and w is the week span. Note that d and w are actually two different types of periods. Comprehensively, d is equal to one-day that

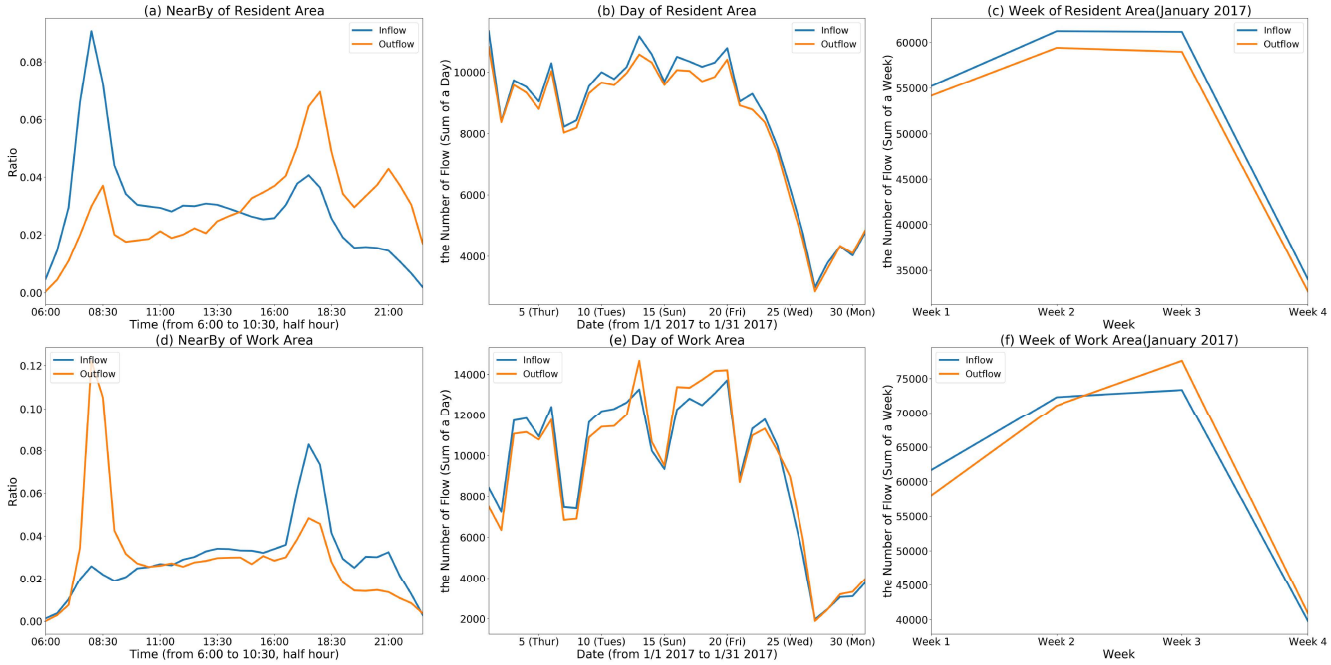

Fig. 5. Temporal influences (Work Area and Residential Area are targeted in Fig. 3).

depicts daily periodicity, and w is equal to weekly that draws the one-week tendency.

E. External Elements

Traffic flows can be affected by many complex external factors, such as weather and event. Fig. 6 (a) and (e) shows that the effect of moderate rain obviously increase the inflow at the peak time in work and resident area. Fig. 6 (b) and (f) shows that the influence of low temperature increase the crowd inflow at at both area with the same pattern of moderate rain. Fig. 6 (c) and (g) shows that crowd inflow during Spring Festival can be significantly different from the inflow during normal day at all day time slots. Fig. 6 (d) and (h) shows that Saturday or Sunday has no evidently peak, at resident area morning peak at 8:30pm changing into slow growth for not going to work in morning and at work area as the minority working with no evening peak at 5:30pm. At each comparison, we try to ensure that other factors, showing at each sub figures' legend, are the same.

Let E_t be the extern feature tensor that denotes all external factors at predicted time slot t. In this model (ST-DRN), we primarily concern holiday, weather, and metadata (Workday and Weekends). The details are introduced in Table II. To predict flows at time t, the holiday and metadata can be straightly obtained by checking the calendar. But the weather at future time slot t is unknown. Usually, one can use the forecasting weather at time slot t or the close weather at time slot $t-1$. Standardly, we employ two fully-connected layers on E_t, the first layer is an embedding layer for each sub-factor followed by an activation. The second layer is applied to map low to high dimensions that have the same shape as X_t. The output of the external component in Fig. 3 is denoted as X_{Ex} with the parameters θ_{Ex}.

F. Fusion

At first, we aggregate the first three parts [16] (i.e. nearby, day, week) of Fig. 1 as follows:

$$X_{RU} = W_n \otimes X_n^{(L+2)} + W_d \otimes X_d^{(L+2)} + W_w \otimes X_w^{(L+2)} \quad (4)$$

where $\otimes$ denotes Hadamard product (i.e., element-wise multiplication), W_n, W_d and W_w are the learnable parameters that adjust the degrees influenced by *nearby*, *day* and *week*, respectively.

Then, we here straightly fuse the output of the first three parts with that of the external part, as shown in Fig. 1. Finally, the predicted value at the tth time slot, denoted by $X_t^{\#}$, is defined as:

$$X_t^{\#} = tanh\left(X_{RU} + X_{Ex}\right) \quad (5)$$

where tanh, the activation function, is a hyperbolic tangent that guarantees values of output are at $[-1, 1]$.

Our ST-DRN will be trained to predict X_t from three sequences of flow tensors and external factor features by minimizing mean squared error between the predicted flow matrix and the true flow matrix:

$$\alpha(\theta) = \left\| X_t^{\#} - X_t \right\|_2^2 \quad (6)$$

where represents all learnable parameters in the ST-DRN.

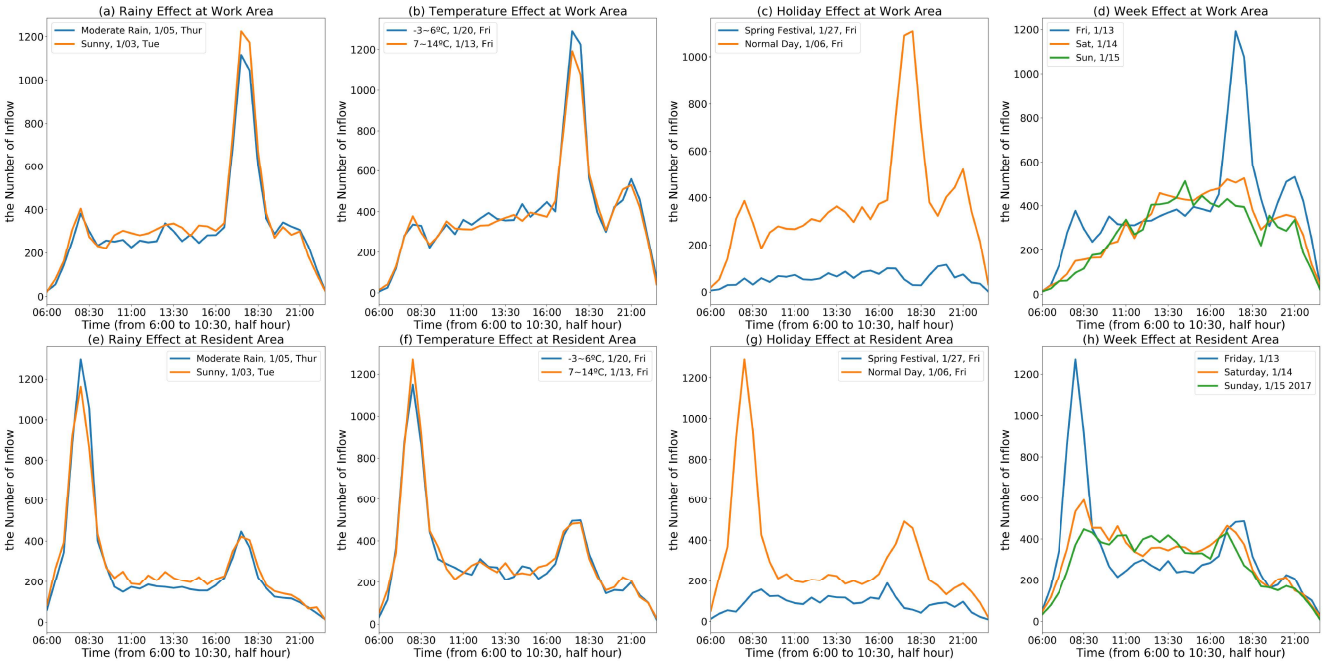

Fig. 6. Influences of holidays, weather and weekend in Work Area and Resident Area (the area is targeted in Fig. 3).

G. Algorithm and Optimization

Algorithm 1 outlines the process of ST-DRN training. We first establish the training instances from the original sequence data. Then, ST-DRN is trained by back-propagation (BP) and Adam [17].

Algorithm 1 ST-DRN Training Algorithm

Input:
Lengths of nearby, day, week sequences: l_n, l_d, l_w;
Day period: d; week period: w;
External factors: $\{E_0, \ldots, E_{n-1}\}$;
Historical data: $\{X_0, \ldots, X_{n-1}\}$;
Output:
Learned ST-DRN model

// step1: establish training instances
$M \leftarrow \phi$
for all *all time slots* t $(1 \leq t \leq (n-1))$ **do**
  $D_n = \left[X_{t-l_n}, X_{t-(l_n-1)}, \ldots, X_{t-1}\right]$
  $D_d = \left[X_{t-l_d.d}, X_{t-(l_d-1).d}, \ldots, X_{t-d}\right]$
  $D_w = \left[X_{t-l_w.w}, X_{t-(l_w-1).w}, \ldots, X_{t-w}\right]$
  // X_t is the target at time slot t
  place an training instance $(\{D_n, D_d, D_w, E_t\}, X_t)$ into M
end for
//step2: training model
Init all learnable parameters θ in ST-DRN
repeat
  randomly select a batch of instances M_t from M
  find θ by minimizing the formula in (6) with M_t
until *stopping criteria is met*

IV. Evaluation

We use the record data collected from subway AFC system of Suzhou in time span from 1st September. 2016 to 31th January. 2017. And we process this data into inflow and outflow 30 minutes as a time slot from 6:00 to 22:30 for each station at every day. Then we use the last month's data to test ST-DRN model and the remaining to train this model. Again, we use this way to train and test our baseline models. We use the ARIMA and SARIMA model to compare ST-DRN.

A. Baselines

ARIMA: Auto-Regressive Integrated Moving Average (ARIMA) [3] is a well-known model for understanding and predicting future values, one of the popular linear models in time series forecasting during the past three decades (reference). Although ARIMA models are extremely, representing several different types of time series, i.e., autoregressive (AR), moving average (MA) and combined AR and MA (ARMA) series. P, Q, D are 3, 34, 2 respectively representing the order of AR, MA and Difference.

SARIMA: Seasonal ARIMA [18]. A seasonal ARIMA model is formed by including additional seasonal terms in the ARIMA models we have seen so far. Seasonality in a time series is a regular pattern of changes that repeats over a regular time periods. S is 3 representing the order of seasonal character adding from ARIMA model.

ANN: It first extracts the temporal features in each station,

and feed into an artificial neural network [19], applying just to pattern the relationship between past and current to forecast the future flows with the same hyper parameters installed in ST-DRN.

B. Preprocessing

We use tanh as our final activation in the output of the ST-DRN in Fig. 1, whose range is between -1 and 1. Hence, we transform the flows data into the range $[-1, 1]$ by the Min-Max normalization method. In the evaluation, we shift the predicted value back to the normal values, compared with the truth flows data. For external factors, we transform weekends or workday, holidays and weather conditions into binary tensors by one-hot code, and use Min-Max normalization to map the Temperature and Wind speed into the range $[0, 1]$.

C. Hyper parameters

The python libraries, including Keras and Theano, are used to build our models. The convolutions of Conv1 and all residual units use 64 filters of size 59×1 (one-dimensional), and Conv2 uses a convolution with 2 filters of size 59×1. The batch size and learning rate are 32, 0.0001 respectively. We partition 80% of the training data for training model, and the remaining is chosen as the validation set, which is used to early-stop our training algorithm for each model based on the best validation score. Afterwards, we continue to train the model on the full training data for a fixed number of epochs (e.g., 150 epochs). There are 5 extra parameters in our model, of which p and q are empirically fixed to the number of time slots at one-day and one-week, respectively. For choices of the three dependent sequences, we propose them as: $l_n \in \{3, 4, 5\}$, $l_d \in \{1, 2, 3, 4\}$, $l_w \in \{1, 2, 3, 4\}$.

TABLE II
DATA SETS (AFC RECORD DATA , HOLIDAYS, WEEKENDS AND WEATHER CONDITIONS)

Data set		**Subway**	
Data type		Subway AFC record data	
Location		Suzhou	
Time span		From 9/1 2016 to 1/31 2017	
Length of time slot		30 minutes	
The number of stations		59	
Statistic information of flows for each station at a time slot on January 2017			
Inflow	max: 2536	mean: 136	std: 167.906
Outflow	max: 3113	mean: 137	std: 170.365
Statistic information of external data			
Holidays		13 days	
Weekends		44 days	
The number of weather		5 types (e.g. Sunny, Rainy)	
Temperature (C°)		$[-3, 34]$	

Evaluation Metric: We measure our method by Root Mean Square Error (RMSE) as:

$$RMSE = \sqrt{\frac{1}{a} \cdot \sum_i (X_i - X_i^{\#})^2} \tag{7}$$

where $x^{\#}$ and x are the predicted value and truth data, respectively; a denotes the number of all predicted values.

D. Results on AFC Data

As shown in Fig. 7, We first give the comparison with baseline models on Suzhou. We provide several variants of ST-DRN with different layers and different factors. Taking L4 for example, it defaulted contains external factors, 40 length of one-dimensional convolution kernel, and has 4 residual units, each of which is consisted of two convolutional layers and before ReLU with batch normalization (BN) as Fig. 2. As default, the three dependent sequences are $\{3, 1, 1\}$ with the order of l_n, l_d, l_w. Comparing with the previous baseline models and the different types of ST-DRN, L4 reduces error to 27.809, which perfectly improves accuracy. ST-DRN has relatively from 34.29% up to 58.16% lower RMSE than three baselines, demonstrating that our proposed model has good performance on flow prediction tasks. However, L4+A1 or L4+A2, using early an hour data before predicted time t, has approximative RMSE with these baselines, demonstrating that our model has not perfectly performance on Non real-time system and inspiring us to modify the model in future. Respectively, the RMSE values of ANN, ARIMA and SARIMA are 55.401, 66.473, 63.130.

- **Number of residual unit:** Results of L2, L4 show that RMSE decreases as the number of residual units increases. On theory, the effects of result increase with adding residual unit layer and we need more data to feed the deeper network, using residual learning. However L6 show poor effect than L4, as the learning rate and batch size same in two experiment or the limited data. Respectively, the RMSE values of L2 and L6 are 31.529, 34.461.
- **Distinct structure of residual unit:** We try some different types of residual units. L4 adopts the standard Residual Unit without special note (see Fig. 2). Compared with L4, Residual Unit of L4+SingleRU only contains 1 ReLU followed by 1 convolution and 1 BN before ReLU, and Residual Unit of L4+NoBN remove two batch normalization layers, each of which is placed before ReLU. We observe that L4+NoBN is worse than L4, and L4+NoBN is better than L4+SingleRU, demonstrating the effectiveness of batch normalization and more convolution, respectively. Separately, the RMSE values of L4+SingleRU and L4+NoBN are 33.079, 31.123.
- **External elements:** L4 considers the external factors, including weekends, holiday and weather. If not, the model is denoted as L4+NoE. The results, that L4 is better than L4+NoE, reveal that external factors are always advantageous.
- **Parametric matrix fusion:** Being different with L4, L4+NoFusion do not use parametric matrix fusion (see Eq. 4). Instead, L4+NoFusion use a straightforward method for fusing, i.e., $X_n^{(L+2)} + X_d^{(L+2)} + X_w^{(L+2)}$. It shows the error greatly increases, which evidences the effectiveness of our used parametric matrix fusion. Respectively, the RMSE values of L4+NoE and L4+NoFusion are 32.751, 30.219.

- **size of convolution kernel:** As we thoroughly discuss in subsection Spatio Components in section III, the different size of convolution kernel has different effect on this data and the best size is around 40, decided by the layout of metro stations, the distribution of ride time length, and the metro stations numbering sequence. Respectively, the RMSE values of L4+10, L4+35 and L4+50 are 36.407, 30.795, 33.963.
- **Early data:** Mentioned at Introduction, we adjust the parameters for ST-DRN to attempt using early one hour data. L4+A1 using $\{l_n, l_d, l_w\}$ of $\{1, 3, 1\}$ show better performs than L4+A2 setting $\{l_n, l_d, l_w\}$ of $\{3, 1, 1\}$ and baseline models in Table III. And the results reveal our model's practical value and guide us to modify ST-DRN at actually using in subway system in the future. Respectively, the RMSE values of L4+A1 and L4+A2 are 58.964, 63.692.

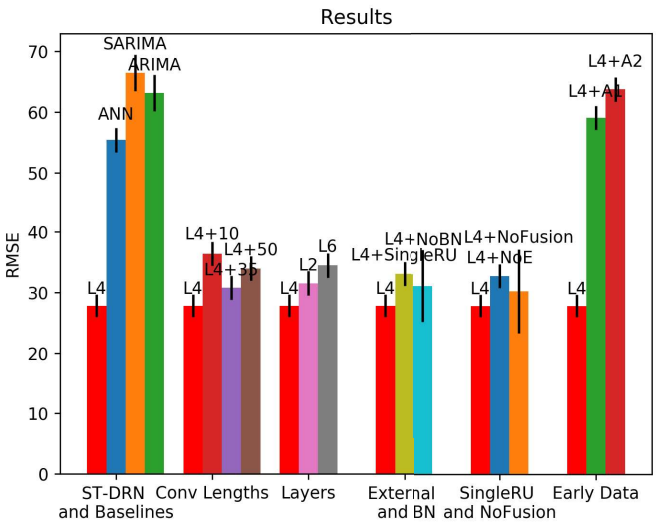

Fig. 7. Results on AFC Data

E. Discussions

In above results, we find that L4, the best forecasting model we proposed, is promoted than ARIMA [3] by 55.94% and L4+A1 is little better than AIMAR in accuracy. Hence, our model make a great improvement in forecasting accuracy with fast training model time using GPU, as the layer of deep residual network is light, if AFC data can be real time collected by system.

V. Conclusion

In this paper, to pattern particular spatio-temporal forecast validly [20], we mainly propose applying the residual learning, inserting CNNs in residual unit in Fig. 2 to handle spatial issue, and a parametric matrix fusion mechanism to integrate the three sub-factors of temporal. We use deep residual learning model for forecasting the flow of crowds in each subway station, based on historical data, weather, weekends and holidays. We evaluate our model on two types of crowd flows in Suzhou AFC record data, attaining significantly performances than 3 baseline methods, making sure that our model is better and more applicable to the flow prediction on subway station.

References

[1] A. Abadi, T. Rajabioun, and P. A. Ioannou, "Traffic flow prediction for road transportation networks with limited traffic data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 653–662, 2015.

[2] N. Lathia and L. Capra, "How smart is your smartcard?: measuring travel behaviours, perceptions, and incentives," in *Proceedings of the 13th international conference on Ubiquitous computing*. ACM, 2011, pp. 291–300.

[3] M. Van Der Voort, M. Dougherty, and S. Watson, "Combining kohonen maps with arima time series models to forecast traffic flow," *Transportation Research Part C: Emerging Technologies*, vol. 4, no. 5, pp. 307–318, 1996.

[4] D. Zhang, R. Jiang, S. Wang, Y. Zhu, B. Yang, J. Cao, F. Zhang, and T. He, "Everyonecounts: Data-driven digital advertising with uncertain demand model in metro networks," in *Big Data (Big Data), 2015 IEEE International Conference on*. IEEE, 2015, pp. 898–907.

[5] D. Zhang, J. Zhao, F. Zhang, R. Jiang, and T. He, "Feeder: supporting last-mile transit with extreme-scale urban infrastructure data," in *Proceedings of the 14th International Conference on Information Processing in Sensor Networks*. ACM, 2015, pp. 226–237.

[6] S. Sun, C. Zhang, and G. Yu, "A bayesian network approach to traffic flow forecasting," *IEEE Transactions on intelligent transportation systems*, vol. 7, no. 1, pp. 124–132, 2006.

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

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[9] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[10] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European Conference on Computer Vision*. Springer, 2016, pp. 630–645.

[12] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[13] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, 2015, pp. 448–456.

[14] M. Mathieu, C. Couprie, and Y. LeCun, "Deep multi-scale video prediction beyond mean square error," *arXiv preprint arXiv:1511.05440*, 2015.

[15] V. Jain, J. F. Murray, F. Roth, S. Turaga, V. Zhigulin, K. L. Briggman, M. N. Helmstaedter, W. Denk, and H. S. Seung, "Supervised learning of image restoration with convolutional networks," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. IEEE, 2007, pp. 1–8.

[16] Y. Zheng, "Methodologies for cross-domain data fusion: An overview," *IEEE transactions on big data*, vol. 1, no. 1, pp. 16–34, 2015.

[17] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[18] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results," *Journal of transportation engineering*, vol. 129, no. 6, pp. 664–672, 2003.

[19] D. Park, L. R. Rilett, and G. Han, "Spectral basis neural networks for real-time travel time forecasting," *Journal of Transportation Engineering*, vol. 125, no. 6, pp. 515–523, 1999.

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