



Article

# Large-Scale Station-Level Crowd Flow Forecast with ST-Unet

Yirong Zhou , Hao Chen, Jun Li <sup>\*</sup>, Ye Wu, Jiangjiang Wu and Luo Chen 

College of Electronic Science, National University of Defense Technology, Changsha 410073, China; zhouyirong09@nudt.edu.cn (Y.Z.); hchen@nudt.edu.cn (H.C.); yewugfd@nudt.edu.cn (Y.W.); wujiangjiang08@nudt.edu.cn (J.W.); luochen@nudt.edu.cn (L.C.)

\* Correspondence: junli@nudt.edu.cn; Tel.: +86-136-3742-4834

Received: 24 January 2019; Accepted: 11 March 2019; Published: 13 March 2019



**Abstract:** High crowd mobility is a characteristic of transportation hubs such as metro/bus/bike stations in cities worldwide. Forecasting the crowd flow for such places, known as station-level crowd flow forecast (SLCFF) in this paper, would have many benefits, for example traffic management and public safety. Concretely, SLCFF predicts the number of people that will arrive at or depart from stations in a given period. However, one challenge is that the crowd flows across hundreds of stations irregularly scattered throughout a city are affected by complicated spatio-temporal events. Additionally, some external factors such as weather conditions or holidays may change the crowd flow tremendously. In this paper, a spatio-temporal U-shape network model (ST-Unet) for SLCFF is proposed. It is a neural network-based multi-output regression model, handling hundreds of target variables, i.e., all stations' in and out flows. ST-Unet emphasizes stations' spatial dependence by integrating the crowd flow information from neighboring stations and the cluster it belongs to after hierarchical clustering. It learns the temporal dependence by modeling the temporal closeness, period, and trend of crowd flows. With proper modifications on the network structure, ST-Unet is easily trained and has reliable convergency. Experiments on four real-world datasets were carried out to verify the proposed method's performance and the results show that ST-Unet outperforms seven baselines in terms of SLCFF.

**Keywords:** crowd flow forecast; irregular grid data; multi-output regression; deep neural network; spatio-temporal analysis

## 1. Introduction

To be able to forecast crowd flow is of great importance for risk assessment and public safety [1,2]; there has been increased emphasis on this since accidents such as the 2014 Shanghai Stampede occurred. Compared with doing citywide or regional forecasts, a station-level crowd flow forecast (SLCFF) benefits public safety protection at the station-level when predicting the flow at those places with high crowd mobility, such as metro/bus/bike stations. Stations are scattered throughout a city and the variation of crowd flow reflects people's daily life: work, activities, home, etc. However, SLCFF can benefit many other applications too, such as traffic management, taxi dispatching, bike-sharing pre-reallocation, etc. Concretely, SLCFF predicts the number of people that will arrive at or depart from stations in a given period.

There are many stations in a city. The crowd flow in one individual station exhibits greater fluctuation than that observed on the cluster level. The crowd flow variation in a station generally complies to the trend at the cluster it belongs to when hierarchical clustering of geo-neighboring stations is applied. The spatial dependence of crowd flow lies in the hierarchical structure of stations. Moreover, the peak arrival crowd at a certain station may have come from several other stations a

while before; and the peak departure would cause fluctuation at nearby or far away stations a while later. Viewing crowd flow at each time slice individually would not reflect the inherent temporal dependence. Furthermore, some external factors, such as weather conditions and events, may change the crowd flow tremendously. All these issues integrally make it challenging to do a station-level crowd flow forecast with high precision. The forecast performance can only be improved when the spatio-temporal dependence and the external factors are well modeled.

Crowd flow forecast is intrinsically a regression problem. From the view of models and the forecasting techniques adopted, the works could be categorized into two groups: one uses empirical statistical methods [3,4] or pattern mining to identify crowd flow hot-spots or activity patterns [5,6]; the other implements machine learning techniques to forecast crowd flow. The former is used to answer when/where/how the future hot-spots might be from a macro perspective. The latter is used to make predictions numerically by modeling the impact factors as much as possible. This paper focuses on the latter.

For a regression problem, from the view of how the many target variables are modeled, the machine learning techniques can be summarized as being single-output models and multi-output models [7]. The former trains the models for each target station individually or just one single-output model, in which the loss function has only one target variable. The latter builds one multi-output model to forecast many real-value target variables, which are optimized jointly in the loss function. Here are some examples of the former type: support vector regression (SVR) methods for traffic flow predictions [8–10], gradient boosting regression tree (GBRT) and multi-similarity-based inference models for bike-sharing demand forecasting [11,12], ensemble framework with time-varying Poisson models and the auto-regressive integrated moving average (ARIMA) model for taxi-passenger demand forecasting [13]. For multi-output models, some examples include: the probabilistic graphical models (PGM)-based hybrid framework for citywide traffic volume estimation [14], intrinsic Gaussian Markov random field (IGMRF) model, one of the PGM models with cluster-based adjustment for cluster-level crowd flow forecast [1], vector auto-regressive moving average (VARMA) with a spatio-temporal correlations matrix for real-time traffic predictions [15],  $\nu$ -SVR (the modified multi-output SVR (M-SVR) method) for traffic speed predictions in large road networks [16], deep spatio-temporal residual networks (with convolutional neural network (CNNs) as kernels) for region-level crowd flow predictions [2], and multi-graph convolutional networks for station-level bike flow predictions [17].

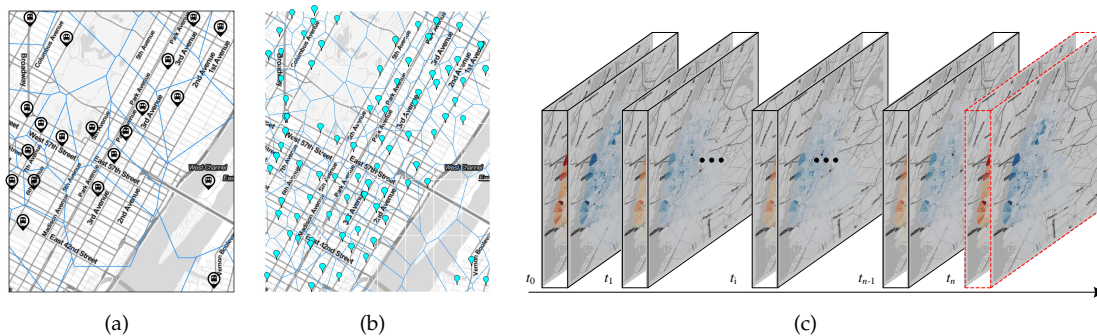
Theoretically, by modeling the relationships between the target variables and optimizing accordingly, multi-output models can guarantee a better representation and interpretability of the real-world problems than single-output models [7], as is shown in the works enumerated. However, many multi-output models (PGMs, M-SVR, VARMA, as mentioned above) exhibit high computational complexity and can not handle large-scale problems (hundreds of target variables) well [7]. Because they model the spatio-temporal dependence of targets carefully, the number of training parameters is often  $k$  times the product of the amount of features and the amount of target variables. To reduce complexity, target variables are grouped by cluster algorithms [16] or part of training parameters are set according to rules (like in Reference [15]), which sacrifices some forecast performance. With abundant designs of structures and mature training techniques to handle large-scale problems well, deep neural networks (DNNs) are currently subject to much research (References [2,17,18], as mentioned above). However, the geo-factors and information about the city are either lost or the forecast is only applicable regionally, because regular grids are leveraged and spatio-temporal dependences are simplified to enable the application of widely used neural networks (CNN, LSTM, etc.).

Inspired by the trend of leveraging DNNs on such large-scale regression problems, we forecast station-level crowd flow with a spatio-temporal U-shape network (ST-Unet) in this paper. It is a neural network-based multi-output regression model, handling hundreds of target variables. Its structure is carefully designed to emphasize stations' local-global dependence of crowd flow to improve the forecast performance. Concretely, the contributions we make in this paper are: (1) gConv-layer (convolutional layer) is designed to handle stations' irregular distribution and learn the influence

of crowd flow from neighbor stations, which is based on the idea of receptive fields and sharing weights from CNNs; (2) the hierarchical information of the stations is integrated into the networks by gUpsampling/gDownSampling-layers, which enhances the model's ability to understand the local-global information of crowd flow; (3) several modifications on the widely used Unet are made, which improve the model's convergency and can handle hundreds of target variables well. Experiments on four real-world datasets were carried out to verify the proposed method's performance. Results show that ST-Unet outperforms seven baselines on station-level crowd forecasting.

## 2. Overview

As shown in Figure 1a,b, stations are irregularly scattered throughout a city. The in-out flow at each station reflects people's mobility level of the located region at the time. Visualized on maps and sequenced through time, we can get a series of double-channel (in-out channels) heat maps (as shown in Figure 1c). Thus, we model the station-level forecast problem by generating a subsequent heat map based on the ordered series of heat maps. Ideas from CNNs and multi source data are utilized to improve the forecast performance. In this section, the formal definition of the SLCFF problem and some preliminaries used are first introduced, and the framework of our method is illustrated as follows.



**Figure 1.** (a,b) Metro stations and bike-sharing stations in New York City; each station serves a region of the city, which could be roughly calculated by Voronoi-based segmentation; (c) the in-out crowd flow at each station aggregated along time axis can be viewed as a series of double-channel heat maps.

### 2.1. Preliminaries & Problem Definition

**Definition 1.** *Stations.* There are  $m$  stations in the station-level crowd flow forecast problem,  $S_i$  ( $i \in [0, m-1]$ ) is the  $i$ th station.

**Definition 2.** *Trip.* A trip  $Tr = (S_o, S_d, t_o, t_d)$  is a record, where  $S_o, S_d$  denote the origin and destination station, respectively,  $t_o$  and  $t_d$  are the timestamps when people depart from  $S_o$  and arrive at  $S_d$ , respectively.

**Definition 3.** *Observing time unit.*  $\tau$  is the observing time unit for aggregating the in-out flow count, e.g., 30 min or 1 h. Let  $T = [\tau_0, \tau_1, \dots, \tau_i, \dots, \tau_{n-1}]$  is the whole observing time period.

**Definition 4.** *In-out flow.*  $\mathbf{x}_{\tau_i}^{out} = [x_{S_0}, x_{S_1}, \dots, x_{S_{m-1}}]_{\tau_i}$  records each station's out-flow count during the time period  $\tau_i$ . Similarly,  $\mathbf{x}_{\tau_i}^{in}$  records the in-flow count. Let  $\mathbf{x}_{\tau_i} = [\mathbf{x}_{\tau_i}^{out} | \mathbf{x}_{\tau_i}^{in}] \in \mathbb{N}^{1 \times 2m}$  concatenate  $\mathbf{x}_{\tau_i}^{out}, \mathbf{x}_{\tau_i}^{in}$  as one record.

**Problem:** The station-level crowd flow forecast problem. Given the historical observations  $\{\mathbf{x}_{\tau_i} | i \in [0, 1, \dots, n-1]\}$ , forecast  $\hat{\mathbf{x}}_{\tau_n}$ , aiming to minimize  $|\hat{\mathbf{x}}_{\tau_n} - \mathbf{x}_{\tau_n}|$ , where  $\mathbf{x}_{\tau_n}$  is the ground truth at  $\tau_n$ .

## 2.2. Framework

Figure 2 shows the framework of our method. To guarantee the performance, multi-source data are adopted, including the in–out records data at each station, locations of stations, road network, meteorology, etc. The ST-Unet model is the forecast model. It merges three Unet branches to capture the temporal dependence of crowd flow and one branch to integrate external factors (see Section 3.1). Each Unet deals with the spatial dependence of crowd flow among the stations (see Section 3.2). Owing to the stations’ irregular distribution, we redesign the receptive field of each station and bring in the hierarchical information of the stations in the Unet.

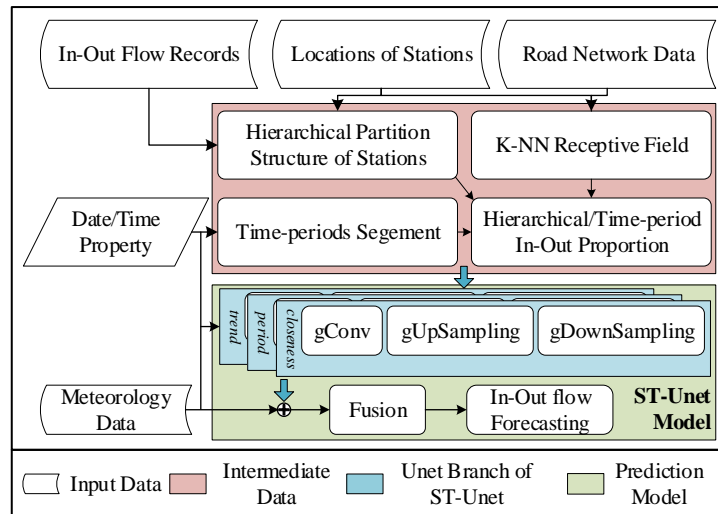
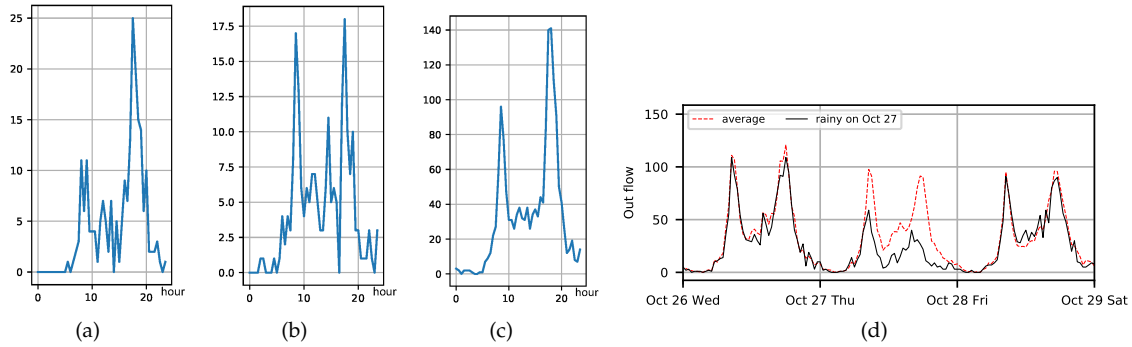


Figure 2. The framework of the forecast model.

- *k*-NN (nearest neighbor) Receptive Field of Each Station. The receptive field of CNNs of each entry in regular grid data is its 8 or 24 neighbor grids (when using  $3 \times 3$  or  $5 \times 5$  feature maps, respectively). However, because the stations are scattered irregularly, the *k*-NN receptive field of each station should be redefined. Inspired by graph-CNNs utilizing graph labelings to impose an order on nodes [19], we define and figure out each station’s receptive field with its *k* ordered nearest neighbor stations reachable in the road network (See Section 3.3).
- *Hierarchical Structure of Stations*. From the view of one individual station, the changing regularity of in–out flow is difficult to determine because of its fluctuation, as shown in Figure 3a,b. However, it is much more robust and regular from the view of a region with several stations, as shown in Figure 3. Thus, we employ an agglomerative clustering algorithm to construct the hierarchical structure of the stations, which is based on the stations’ geo-locations and historical in–out flow data. This is used as auxiliary information to determine the ‘pools’ of downsampling/upsampling layers in ST-Unet, which enhance the forecasting stability (See Section 3.4).
- *Time-periods Segmentation*. Considering the temporal heterogeneity of crowd flow, we categorize according to seven time periods: 1. 7:00 a.m.–11:00 a.m. (morning rush hours); 2. 11:00 a.m.–4:00 p.m. (day hours); 3. 4:00 p.m.–9:00 p.m. (evening rush hours); and 4. 9:00 p.m.–7:00am (night hours); and 5. 0:00 a.m.–9:00 a.m. (night hours); 6. 9:00 a.m.–7:00 p.m. (trip hours); and 7. 7:00 p.m.–12:00 p.m. (evening hours) on weekends/holidays. Each time slot  $\tau_i$  is labelled with a property field ‘*hd*’ indicating what kind of time periods it belongs to, i.e.,  $\tau_i.hd \in [1, 7]$ .
- *Hierarchical/Time-period In–out proportion*. According to the hierarchical structure of stations and different time periods, the maximum likelihood estimation method is used to estimate each station’s in–out flow proportion within its cluster. Such information is used to correct the up-sampling operation, replacing the usual adopted method—padding (See Section 3.4).

Except for the intermediate data above, external features should also be prepared, including date/time properties and weather. Date/time properties include weekday/weekend, holiday, time slot, and so on. Weather conditions affect the crowd flow in some degree, as shown in Figure 3d. Weather features include precipitation, visibility, and so on. With these intermediate data and external data, ST-Unet can be trained to predict each station's crowd flow in a given period. Its architecture is elaborated in Section 3.1.



**Figure 3.** (a,b) The variation in out-flow count of two stations in one day. (c) The variation in out-flow count of a region with seven stations in one day. (d) The effect of weather on the fluctuation of crowd flow (27 October, 2016 in New York City was a rainy day).

### 3. ST-Unet

#### 3.1. Overview

Figure 4 presents the architecture of ST-Unet. It is composed of four branches: (a) Three Unet branches respectively capture the temporal influence, crowd flow of closeness (related to the recent time period), period (yesterday for the same time period), trend (last week during the same time period); each branch is a modified Unet capturing the spatial dependence of crowd flow, illustrated in Section 3.2. (b) One branch introduces external influence, which contains weather and date/time property features in this paper.

As illustrated in Definition 4, we use only 1-d vector to present the double-channel heat map of crowd flow in Figure 1c, i.e., the in-flow and out-flow of all stations during one time slot, which simplifies the subsequent operations. Then, we stack specified in-out flow records of different time slots to capture the variation along the time axis in the three colored branches:

$$\begin{aligned} \mathbf{X}_c &= [\mathbf{x}_{n-l-1}^T, \dots, \mathbf{x}_{n-1}^T]^T \\ \mathbf{X}_p &= [\mathbf{x}_{n-l-\tau_p}^T, \dots, \mathbf{x}_{n-\tau_p}^T]^T \\ \mathbf{X}_t &= [\mathbf{x}_{n-l-\tau_t}^T, \dots, \mathbf{x}_{n-\tau_t}^T]^T \end{aligned}$$

where  $l$  is the length of time slots chosen to stack,  $l \geq 1$ ,  $\tau_p$  and  $\tau_t$  are the length of one day's and one week's time slots, respectively. The blue branch stacks the records of recent time slots; the green branch stacks the records of the same time slots as yesterday; the red branch stacks the records of the same time slots as last week. They separately model three temporal properties: closeness, period, and trend. The bottom branch uses one layer of the fully connected neural network to introduce the external feature vector, including weather and date/time properties. The weather features include precipitation, wind speed, temperature, visibility, etc. The date/time property features contain workday/weekend, holiday or not, kind of time period, etc.

### 3.2. Unet Branch of ST-Unet

The Unet branch of ST-Unet is thus named because of its U-like network shape, as shown in Figure 5. It is inspired by the widely used network Unet in the domain of medical image processing [20]. It is usually applied in pixel-level image segmentation, i.e., to classify each pixel in an image. The horizontal architecture of Unet performs CNNs on different hierarchical resolutions of the image. The design emphasizes the local-global dependence of the entire image on the output.

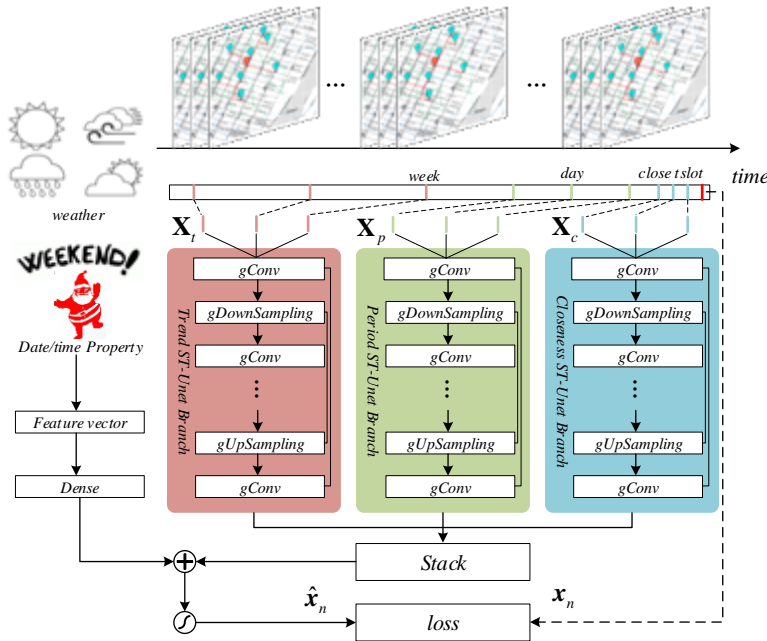


Figure 4. The architecture of the spatio-temporal U-shape network (ST-Unet).

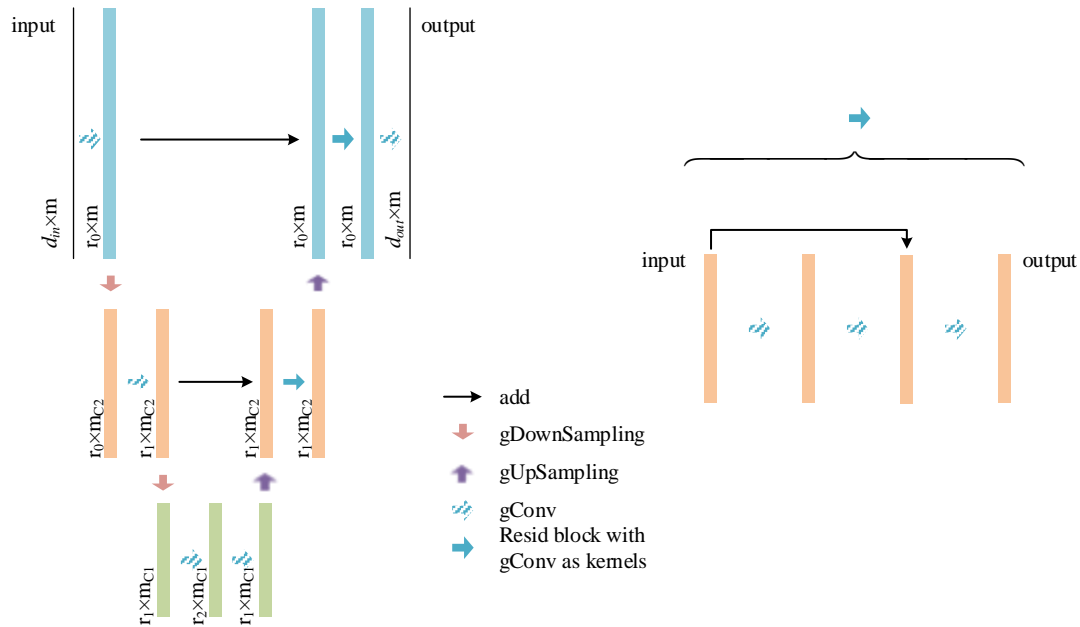


Figure 5. On the left is the architecture of a single Unet branch; on the right is the unfolded format of the resid block with gConv as kernels.

The architecture is well-suited to SLCCF, especially when the multi-output model is adopted and there are hundreds of stations. Local and global features both exist in crowd flow in urban



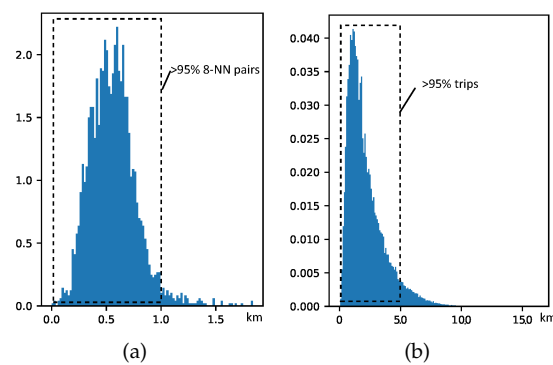
areas. The local features are a result of the distribution of points of interest and different regions' functionality. The global features are a consequence of different time periods of the day, weather, or events. Combining local and global information related to crowd flows at stations enhances the station-level forecast.

In Figure 5, *gConv*, *gDownSampling*, and *gUpSampling* are designed to deal with the irregular distribution of the stations, respectively, similar to convolutional layer, downsampling layer, upsampling layer in CNNs. They are elaborated in Sections 3.3 and 3.4. The horizontal architecture uses convolutional operations, with two 'bridges' in the two shallower layers.  $r_0, r_1, r_2$  are the feature channels' count of *gConv*. There are two downsampling layers and two upsampling layers determined by the hierarchical structure of the stations (see Section 3.4).  $m_{C1}$  and  $m_{C2}$  are the clusters count of the corresponding layer in the hierarchical structure, respectively.

Different from the usage of Unet in pixel-level image segmentation, the outputs in our model are real values, which require higher precision. For that reason, several modifications were tested before being finally adopted. First, the depth of the hierarchical structure is shallow as the 'pixels' are not as numerous as those in images (hundreds of stations compared to  $572 \times 572$  image, as in Reference [20]). Second, the bridges use an 'add' operation as a highway instead of 'concat'. Third, the resid blocks are immediately after the bridges. These modifications make the network more trainable and achieve reliable convergence.

### 3.3. *gConv*

Generally, the site-selection of metro/bus/bike-sharing stations is well-designed and stations are scattered throughout a city. Using New York City Citi Bike as an example, most stations' 8 nearest neighbors (8-NN) are reachable within 1.0 kilometer in the road network (shown in Figure 6a). According to Tobler's first law of geography [21], everything is related to everything else and near things are more related than distant things. That means the crowd flow at a station is probably related to its neighbor stations. This is the insight of the convolutional layers in the shallower layers of the Unet branch. Furthermore, as can be seen in Figure 6b, the distance of most trips does not exceed 5 km and is mainly between 1.5 km and 4 km. That means when people leave an area, they do not generally go too far. This is the insight from the convolutional layers in the deeper layers of the Unet branch.



**Figure 6.** (a) The distance distribution of 8 nearest neighbors (8-NN) stations reachable on the road network of each station. (b) Trip distance distribution in New York Citi Bike-sharing system.

The  $k$ -NN receptive field of each station should be redefined as they are not in regular grids. Inspired by the works of graph-CNNs [19,22], we formalize the convolutional layer *gConv* in this paper as follows.

As shown in Figure 7, a rectangular buffer is first used to roughly determine the neighbors of a station  $S_i$ . Then, the  $k$  nearest neighbors are determined and ordered, according to the road network and using the shortest path between stations to measure the distance. These stations are the receptive

field of station  $S_i$ . The cases in the deeper layers of Unet branch are different, treating each cluster as one station located on its centroid and using Euclidean distance between centroids as measurement.

To simplify the operations, we assume only the prediction of out-flow for a moment. Let  $w = [w_0, w_1, \dots, w_k] \in \mathbb{R}^{1 \times (k+1)}$  is a feature map. As shown in Figure 7, the convolutional operation of each station is  $x \cdot w^T$ . For all stations,  $gConv$  can be depicted in matrix form:

$$gConv(x_{\tau_i}^{out}, \mathbf{WK}) = f(x_{\tau_i}^{out} \cdot \mathbf{WK}), \tag{1}$$

where  $f(\cdot)$  is the activation function,  $x_{\tau_i}^{out}$  records each station's out-flow during time period  $\tau_i$ ,  $\mathbf{WK}$  is the result of filling k-NN matrix  $\mathbf{K}'$  with feature map  $w$ . The k-NN matrix

$$\mathbf{K}' = [K_{ij}], K_{ij} = \begin{cases} p + 1, & S_j \xrightarrow{p^{th}} S_i \\ 1, & i = j \\ 0, & other \end{cases} \tag{2}$$

$S_j \xrightarrow{p^{th}} S_i$  means station  $S_j$  is the  $p^{th}$  nearest station to station  $S_i$ . Thus, filling  $\mathbf{K}'$  with  $w$  to get  $\mathbf{WK}$ , denoted as  $\mathbf{WK} = w \diamond \mathbf{K}'$ , embedding operations intrinsically:

$$\mathbf{WK} = w \diamond \mathbf{K}' = [wk_{ij}]_{m \times m}, wk_{ij} = \begin{cases} w[K_{ij} - 1], & K_{ij} \neq 0 \\ 0, & K_{ij} = 0. \end{cases} \tag{3}$$

In this paper, the crowd flow forecast includes in-flow and out-flow. The operations above are easily extended with slight modifications. First, the feature map  $w$  should be extended as

$$w = \begin{bmatrix} w_{00} & w_{01} \\ w_{10} & w_{11} \end{bmatrix} \in \mathbb{R}^{2 \times 2(k+1)} \tag{4}$$

to learn both the in-out crowd flow patterns. The k-NN matrix is extended as

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}' & \mathbf{K}' \\ \mathbf{K}' & \mathbf{K}' \end{bmatrix}, \tag{5}$$

$\mathbf{WK}$  is extended as

$$\mathbf{WK} = \begin{bmatrix} w_{00} \diamond \mathbf{K}' & w_{01} \diamond \mathbf{K}' \\ w_{10} \diamond \mathbf{K}' & w_{11} \diamond \mathbf{K}' \end{bmatrix}, \tag{6}$$

and

$$gConv(x_{\tau_i}, \mathbf{WK}) = f(x_{\tau_i} \cdot \mathbf{WK}), \tag{7}$$

where  $x_{\tau_i}$  records each station's in-out flow during the time period  $\tau_i$ . The form of  $gConv$  for multi-channels and in the deeper layers of Unet branch is similar.

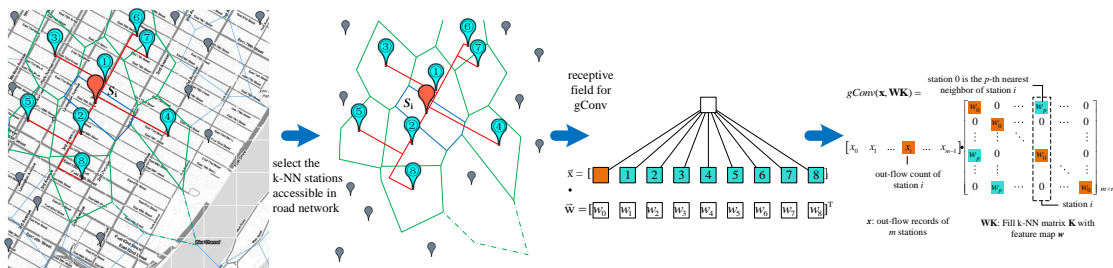


Figure 7. The details of  $gConv$ .



3.4. gDownSampling and gUpSampling

As depicted in Section 3.3, neighbor stations or stations in neighbor areas have highly-related crowd flows. Furthermore, as shown in Figure 3, the periodicity and regularity of a single station’s crowd flow seems chaotic; but it becomes distinguishable from the view of a region with several stations. These insights inspire us to group stations by a hierarchical structure. Such hierarchical structures can be leveraged to determine the ‘pools’ of downsampling/upsampling layers in the Unet for enhancing the forecasting stability. As shown in Figure 8a, the bottom layer is each station itself. The middle layer  $\ell_2$  and the top layer  $\ell_1$  are extracted using an agglomerative clustering algorithm, which is based on the stations’ locations and historical data of stations’ in–out flow data. The historical flow data are used to estimate the transition probability of each station’s in–out flow from/to other stations. Then, the similarity of crowd flow patterns between stations is measured and used as the weighted coefficient.

**Definition 5.** Each station’s feature vector of in–out flow transition probability. Let  $\mathbf{l}_{S_i,hd}=[l_{S_0}^{in}, \dots, l_{S_{m-1}}^{in}, l_{S_0}^{out}, \dots, l_{S_{m-1}}^{out}]_{hd}$  denotes the transition probability of station  $S_i$ ’s in–out flow from/to other stations.  $\sum l_{S_i}^{in} = 1$  and  $\sum l_{S_i}^{out} = 1$ .

$\mathbf{l}_{S_i,hd}$  is estimated using the maximum likelihood estimation method according to each station’s historical in–out flow records. Subscripts  $hd$  are used to distinguish different kind of time periods (see Section 2.2).

**Definition 6.** Similarity of crowd flow between two stations. The cosine of two vectors  $\cos(\mathbf{l}_{S_i,hd}, \mathbf{l}_{S_j,hd})$  is used to measure the similarity between station  $S_i$  and  $S_j$ .

$$\theta(S_i, S_j) = \frac{\cos(\mathbf{l}_{S_i,hd}, \mathbf{l}_{S_j,hd})}{\|\mathbf{l}_{S_i,hd}\| \|\mathbf{l}_{S_j,hd}\|} = \frac{1}{7} \sum_{hd} \frac{\mathbf{l}_{S_i,hd} \cdot \mathbf{l}_{S_j,hd}^T}{\|\mathbf{l}_{S_i,hd}\| \|\mathbf{l}_{S_j,hd}\|} \tag{8}$$

**Definition 7.** Distance matrix. Let  $\mathbf{D} = [d_{ij}]_{m \times m}$  be the distance matrix recording the distance of any two station,  $d_{ij}$  is the shortest path’s length of station  $S_i$  and  $S_j$ . Supposing there are  $m_{C2}$  clusters in the middle layer, let  $\mathbf{D}'=[d'_{ij}]_{m_{C2} \times m_{C2}}$  be the matrix recording the Euclidean distance of any two clusters’ centroids.

**Definition 8.** Stations’ proximity matrix. Let  $\theta = [\theta(S_i, S_j)]_{m \times m}$  be the matrix measuring the similarity of crowd flow patterns of any two stations, and is used as weighted coefficient to point-multiply distance matrix to get stations’ proximity matrix  $\mathbf{Z}$ :

$$\mathbf{Z} = \theta \odot \mathbf{D} \tag{9}$$

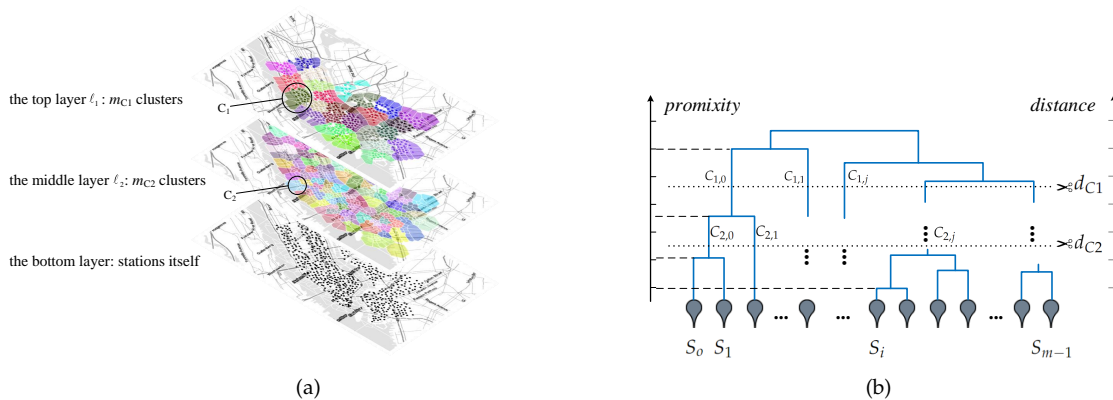


Figure 8. (a) The hierarchical structure of all stations; (b) the tree result from agglomerative clustering.

Figure 8b shows the core idea of agglomerative clustering to determine the hierarchical structure of stations. Agglomerative clustering is a ‘bottom-up’ type hierarchical clustering: each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy [23]. Two goals are achieved by using the stations’ proximity matrix  $\mathbf{Z}$ . First, stations in one cluster should be close to each other in the road network. Second, stations in the same group have similar in-out crowd flow patterns.

As mentioned in Section 3.2, the depth of the hierarchical structure in each Unet branch is shallow. In this paper, the output layers cut from the clustering tree are set to three as shown in Figure 8. Thus, two constraints are used to restrict each cluster’s size on a respective layer: the distance between any two stations in each cluster does not exceed  $d_{C1}$  (for the top layer, constraint  $C_1$ ) and  $d_{C2}$  (for the middle layer, constraint  $C_2$ ), respectively ( $d_{C1} > d_{C2}$ ).

With the hierarchical structure of all stations, the ‘pool’ of the  $gDownSampling/gUpSampling$  layer in the Unet branch can now be determined. Similarly, to simplify the operations, we assume only the out-flow prediction first.

As shown in Figure 9a,b,

$$gDownSampling(x_{\tau_i}^{out}, \mathbf{D}_{C2}) = f(x_{\tau_i}^{out} \cdot \mathbf{D}_{C2}), \tag{10}$$

$$gUpSampling(h_{\tau_i}, \mathbf{U}_{C2}) = f(h_{\tau_i} \cdot \mathbf{U}_{C2}), \tag{11}$$

where  $\mathbf{D}_{C2}$  and  $\mathbf{U}_{C2}$  are determined from the hierarchical structure of all stations, let  $h_{\tau_i}$  be the result of  $gConv$  in the middle layer  $\ell_2$ . Each row in  $\mathbf{D}_{C2}$  and each column in  $\mathbf{U}_{C2}$  correspond to a certain station. Each column in  $\mathbf{D}_{C2}$  and each row in  $\mathbf{U}_{C2}$  correspond to a certain cluster in the middle layer  $\ell_2$ . Concretely,

$$\mathbf{D}_{C2} = [d_{ij}]_{m \times m_{C2}}, d_{ij} = \begin{cases} 1, & S_i \succ C_{2,j} \\ 0, & otherwise \end{cases} \tag{12}$$

$$\mathbf{U}_{C2} = [u_{ji}]_{m_{C2} \times m}, u_{ji} = \begin{cases} u_j^i \neq 1, & S_i \succ C_{2,j} \\ 0, & otherwise \end{cases} \tag{13}$$

where  $S_i \succ C_{2,j}$  means station  $S_i$  belongs to cluster  $C_{2,j}$  in the layer  $\ell_2$ ,  $\sum_{i=0}^{m-1} u_j^i = 1$ ,  $u_j^i$  is the station  $S_i$ ’s out-flow proportion in cluster  $C_{2,j}$ , estimated by the maximum likelihood estimation method according to different time periods. The non-zero entries in each column in  $\mathbf{D}_{C2}$  and in each row of  $\mathbf{U}_{C2}$  denote the ‘pools’ of  $gDownSampling$  and  $gUpSampling$ , respectively.

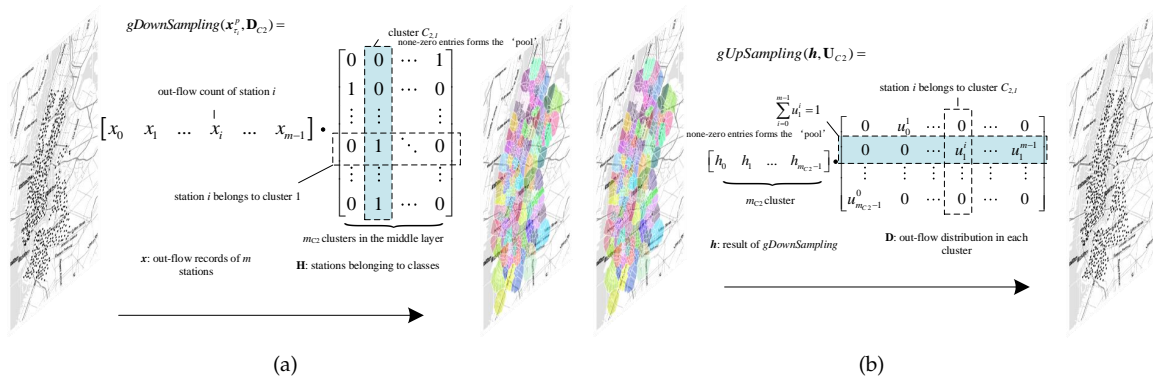


Figure 9. (a)  $gDownSampling$ ; (b)  $gUpSampling$ .

In this paper, station-level crowd flow forecasts includes in and out flow. The operations above are easily extended with slight modifications. First, the ‘pools’  $\mathbf{D}_{C2}$  and  $\mathbf{U}_{C2}$  should be extended as

$$\mathbf{D}_{C2} = \begin{bmatrix} \mathbf{D}_{C2}, & \mathbf{0} \\ \mathbf{0}, & \mathbf{D}_{C2} \end{bmatrix} \in \mathbb{N}^{2m \times 2m_{C2}} \quad (14)$$

$$\mathbf{U}_{C2} = \begin{bmatrix} \mathbf{U}_{C2}, & \mathbf{0} \\ \mathbf{0}, & \mathbf{U}_{C2} \end{bmatrix} \in \mathbb{R}^{2m_{C2} \times 2m} \quad (15)$$

and,

$$gDownSampling(\mathbf{x}_{\tau_i}, \mathbf{D}_{C2}) = f(\mathbf{x}_{\tau_i} \cdot \mathbf{D}_{C2}), \quad (16)$$

$$gUpSampling(\mathbf{h}_{\tau_i}, \mathbf{U}_{C2}) = f(\mathbf{h}_{\tau_i} \cdot \mathbf{U}_{C2}), \quad (17)$$

where  $\mathbf{h}_{\tau_i}$  is the result of  $gConv$  in the middle layer at  $\tau_i$ . The form of  $gDownSampling$  and  $gUpSampling$  for the top layer  $\ell_1$  are similar.

#### 4. Experiments

Experiments to verify ST-Unet’s effectiveness are presented in this section. Three bike-sharing trip datasets and one taxi-trip record dataset were used. All experiments were conducted on a virtual machine with 32 GB RAM and Python 2.7 with tensorflow-1.7.

##### 4.1. Datasets

The three sharing-bike trip datasets are from New York Citi Bike in New York City (<http://www.citibikenyc.com/system-data>), Capital-Bikeshare in Washington DC (<http://www.capitalbikeshare.com/system-data>), and DivvyBikes in Chicago (<http://www.divvybikes.com/data>). The taxi-trip record dataset is the yellow taxi-trip records from NYC Taxi and Limousine Commission (TLC) ([http://www.nyc.gov/html/tlc/html/about/trip\\_record\\_data.shtml](http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml)). They are named **CITI**, **DC**, **DIVVY**, **TAXI** in the following. The details are presented in Table 1.

**Table 1.** Details of the datasets.

Data Source	Citi	DC	Divvy	Taxi
Time Span	1 April–30 October, 2016		1 April–30 October, 2017	
Stations	572	367	469	263 *
Records	9,796,166	2,343,044	2,853,665	65,235,951

\* Since 1 July 2016, only the origin/destination of each taxi trip was recorded; with the taxi zone ID according to the taxi zone map ([https://s3.amazonaws.com/nyc-tlc/misc/taxi\\_zones.zip](https://s3.amazonaws.com/nyc-tlc/misc/taxi_zones.zip)). In the experiments, the centroid of each taxi zone was treated as a location where a station was located.

The meteorology dataset used is from site (<https://mesowest.utah.edu/>) and the selected stations are New York City Central Park (ID: KNYC), Washington (ID: WASD2), Chicago Midway Airport (ID: KMDW). The missing records in the meteorology dataset were filled according to the records from the previous hours. The weather features include: relative-humidity, wind-speed, visibility, sea-level-pressure, precipitation-accumulation (1 h), precipitation-accumulation (3 h), temperature.

Data from 1 April to 19 September were used as training data; data from 19 September to 9 October were used as validating data; and data from 10 October to 30 October were used as testing data. In the testing data, 10 October 2016 was Columbus Day, which was a public holiday. The rainy dates and foggy dates in New York City (NYC), Chicago, and Washington DC are shown in Table 2.

**Table 2.** The rainy dates and foggy dates in New York City, Chicago, and Washington DC.

Cities	Rainy Dates	Foggy Dates
NYC	21/10/2016, 27/10/2016, 29/10/2017	21/10/2016, 27/10/2016, 30/10/2016, 29/10/2017
Chicago	16/10/2016	26/10/2016, 27/10/2016
DC	-	12/10/2016, 13/10/2016, 17/10/2016

Rainy: precipitation-accumulation (1 h) >0.2; foggy: visibility <2.

#### 4.2. Hyperparameters Selection of ST-Unet

The selection of the hyperparameters significantly affects the performance of most deep learning-based models. However, since the training of ST-Unet requires a great deal of time, we present here the best hyperparameters for ST-Unet as well as other settings we tested.

- Observing time unit  $\tau$ : 30 min.
- $k$  nearest neighbor stations: 4.
- The length  $l$  of time slots chosen to stack in the input of each Unet branch: 4.
- Constraints  $d_{C1}$  and  $d_{C2}$  to limit the size of each cluster of layer  $\ell_1, \ell_2$ : 2.5 km and 1.5 km (10.0 km and 5.0 km for **Taxi** dataset).
- $r_0, r_1, r_2$  in each Unet: 16, 16, 16.
- Activation function of  $gConv, gDownSampling, gUpSampling$ : relu.
- Loss function: as the metric *MAE* depicted in Section 4.3.
- Optimizer: Adam-optimizer [24].
- Terminated condition: The training reaches 400 iterations, or when the model does not achieve further improvement for 25 consecutive iterations on the validating data.

#### 4.3. Baselines & Metric

In order to confirm the effectiveness of ST-Unet, we conducted experiments to compare ST-Unet with seven baselines:

- **XGB**: XGB, short for eXtreme Gradient Boosting, is an implementation of GBRT (gradient boosted decision trees) [25]. All input features are the same as ST-Unet.
- **Ensemble**: The ensembles of three predictive models proposed in Reference [13]: ARIMA, time-varying Poisson model, weighted time-varying Poisson model.
- **VARIMA**: Vector-ARIMA extends ARIMA to the multivariate case, which can capture the pairwise relations among the multi-time series.
- **FC**: A three-layers of Full-Connected neural networks is built. Its output is the forecast of all stations' in-out crowd flow. All input features are the same as ST-Unet.
- **MG-CNN**: Multi-graph convolutional networks, a deep neural network model with multiple graphs fusing CNNs for station-level future bike flow forecast [17]. The past six time slots history data are used to forecast the flow in the next time slot.
- **Unet**: Forecasting with only the closeness Unet branch (see Section 3.1).
- **ST-net**: Neither  $gDownSampling$  or  $gUpSampling$  are in the Unet branches, being replaced by  $gConv$ .

**XGB** and **Ensemble** are single-output models, i.e., all stations' forecast models of in or out crowd flow were trained, respectively. **VARIMA**, **FC**, **MG-CNN**, and **ST-Unet** are multi-output models. Owing to the heavy computational costs of **VARIMA**, the second layer  $\ell_2$  of stations was used and **VARIMA** were trained for each cluster, respectively. **Unet** and **ST-net** are the simplified versions used to verify the design of **ST-Unet**.

The metric we adopted to measure the results is *Mean Absolute Error (MAE)*:

$$MAE = \frac{1}{2mT} \sum_{i=1}^T \|\hat{x}_{\tau_i} - x_{\tau_i}\|_1, \quad (18)$$

where  $x_{\tau_i}$  is the ground-truth of all stations' in-out flow, while  $\hat{x}_{\tau_i}$  is the corresponding forecast value.

#### 4.4. Results

Table 3 shows the station-level crowd flow forecast error of ST-Unet compared with the other baselines. The performance of ST-Unet and the seven baselines are measured according to all days (*whole*), different weather conditions (*rainy* and *foggy*), and different time periods (*holidays*, *workdays*, *weekends*). Figure 10 shows some examples of the forecast results of the datasets **CITI** and **TAXI**.

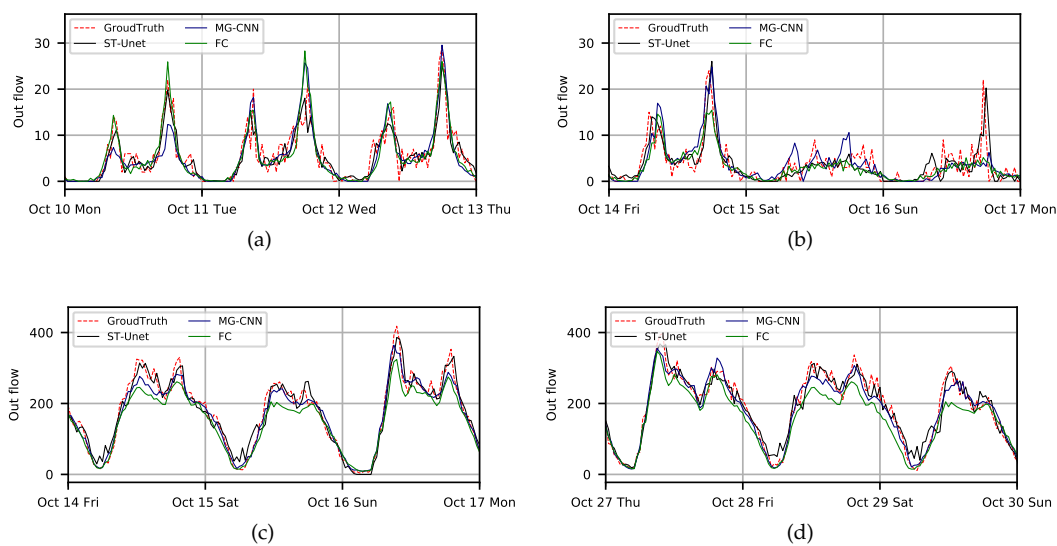
**Table 3.** Station-level crowd forecasting error of ST-Unet compared with the baselines.

Dataset	Methods	XGB	Ensemble	VARIMA	FC	MG-CNN	Unet	ST-net	ST-Unet
<b>CITI</b>	<i>whole</i>	1.057	1.067	1.247	1.077	1.023	1.028	1.103	0.98
	<i>workday</i>	1.088	1.044	1.301	1.126	1.074	1.061	1.145	1.019
	<i>weekend</i>	0.979	1.125	1.112	0.955	0.896	0.946	0.998	0.883
	<i>holiday</i>	1.072	1.07	1.279	1.097	1.006	1.101	1.14	1.046
	<i>rainy</i>	0.98	1.111	0.86	0.974	0.918	0.841	0.897	0.822
	<i>foggy</i>	0.988	1.103	0.913	0.93	0.926	0.887	0.932	0.843
<b>DC</b>	<i>whole</i>	0.489	0.519	0.46	0.497	0.501	0.493	0.472	0.425
	<i>workday</i>	0.481	0.479	0.451	0.487	0.494	0.493	0.471	0.428
	<i>weekend</i>	0.509	0.619	0.483	0.522	0.519	0.493	0.475	0.418
	<i>holiday</i>	0.515	0.487	0.471	0.48	0.505	0.498	0.479	0.419
	<i>rainy</i>	-	-	-	-	-	-	-	-
	<i>foggy</i>	0.499	0.468	0.48	0.512	0.49	0.502	0.482	0.443
<b>DIVVY</b>	<i>whole</i>	0.442	0.448	0.417	0.404	0.422	0.442	0.489	0.39
	<i>workday</i>	0.439	0.43	0.412	0.401	0.411	0.43	0.475	0.392
	<i>weekend</i>	0.45	0.493	0.43	0.412	0.45	0.472	0.524	0.385
	<i>holiday</i>	0.514	0.533	0.494	0.488	0.524	0.545	0.644	0.517
	<i>rainy</i>	0.43	0.438	0.4	0.387	0.433	0.426	0.399	0.364
	<i>foggy</i>	0.342	0.324	0.284	0.298	0.321	0.335	0.318	0.285
<b>TAXI</b>	<i>whole</i>	3.552	3.786	4.372	4.601	3.642	3.731	3.422	3.23
	<i>workday</i>	3.463	3.478	4.287	4.54	3.463	3.613	3.37	3.142
	<i>weekend</i>	3.775	4.556	4.585	4.754	4.09	4.026	3.552	3.45
	<i>holiday</i>	-	-	-	-	-	-	-	-
	<i>rainy</i>	4.411	4.599	4.29	4.684	4.438	4.348	4.081	4.043
	<i>foggy</i>	4.411	4.599	4.29	4.684	4.438	4.348	4.081	4.043

On the whole, ST-Unet performs well on the four datasets, which was expected. Compared with all the baselines, ST-Unet reduces the average forecast error by 10.8% (**CITI**), 15.3% (**DC**), 12.2% (**DIVVY**), 19.8% (**TAXI**), respectively. The much better performance for **TAXI** is due to the large amount of people taking taxis, as shown in Figure 10c,d. The average forecast error reduced against the baselines is around 9.4% for workday instances, while it is 6.1% for weekend instances. Furthermore, as external factors are brought in, ST-Unet also performs well under different weather conditions. The forecast errors are reduced by 6.2% on average. We found that most time periods that coincide with foggy days in the test datasets are in the evening or midnight, such as 21:00–21:20 on 26/10/2016, 23:00 on 26/10/2016, and 5:30 on 27/10/2016 in Chicago. In addition, the precipitation-accumulation (1 h) on rainy days are 0.43 mm on average and 0.54 mm max on 29/10/2017 in New York City. It seems

that the influence that this made on crowd flow was not great, for this reason ST-Unet only performed slightly better than the other methods.

As shown in Table 3, ST-Unet with other deep learning-based methods (FC and MG-CNN) shows better performance than the other baselines on the whole. However, FC shows an unstable forecast performance among different datasets, owing to its fully-connected layers containing too many parameters for training. ST-Unet performs slightly better than MG-CNN. With long short term memory (LSTM) cells, MG-CNN uses only the history data from the past six time slots to forecast the flow in the subsequent time slot [17]. Details can be found in Figure 10, where we present some forecast examples of different time periods (holiday: 10/10/2016; weekends: 15–16/10/2016 and 14–15/10/2017; rainy day: 29/10/2017; the others are workdays). It shows that ST-Unet performs better than FC and MG-CNN at the peaks of crowd flow.



**Figure 10.** (a,b) NYC: Forecast out-flow results for a station [*id*=498, Broadway & W 32 St] from 10/10/2016 to 12/10/2016 and 14/10/2016 to 16/10/2016, where 10/10/2016 is holiday, and 15–16/10/2016 is a weekend; (c,d) TAXI: Forecast out-flow results for a station [*id*=161, Midtown Center] from 14/10/2017 to 16/10/2017 and 27/10/2017 to 29/10/2017, where 14–15/10/2017 is weekend, and 29/10/2017 is a rainy day.

To verify the design of ST-Unet, we present the forecast results of **Unet** and **ST-net** (see Table 3). The former has only a closeness Unet branch and the latter has no *gDownSampling* and *gUpSampling*. The forecast performance was improved by 13.1% on average by ST-Unet, showing the necessity of introducing the period/trend branch and the hierarchical structure of stations. It is worth noting that **ST-net** does not perform better than **Unet**. It is probable that the highways in ST-Unet are quite important for the networks' training and 'add' is better than 'concat' (see Section 3.2), as determined after attempts on multiple network structures.

## 5. Conclusions and Discussion

In this paper, we propose a deep learning-based model, named ST-Unet, to make station-level crowd flow forecasts. We present our methods as regards combining the geographic information with the design of the neural network. Three Unet branches to capture the temporal influence and one branch to introduce external influence were integrated into the forecast model. To deal with the irregular grid format of the data in the Unet branch, we propose *gConv* and *gDownSampling/gUpSampling* to replace the corresponding widely used convolutional layer and downsampling/upsampling layers in CNNs. Specifically, to make *gConv* effective, the receptive



field is determined by each station's k-NN as regards local stations reachable on the road network. To make *gDownSampling/gUpSampling* effective, we regularize the 'pools' according to the hierarchical structure of stations, which is extracted using an agglomerative clustering algorithm based on the stations' locations and the historical flow data. Compared with several baselines, ST-Unet generally performed well in the experiments. ST-Unet accurately predicted each station's in-out flow in a future period.

It is notable that the proposed method does not show much superiority with regards to predictions involving rainy/foggy days and holidays. The reason may be the insufficiency of training instances on special days; however, this requires further study. In addition, the multi-time steps forecast performance of ST-Unet was not explored; how ST-Unet can be modified to do this is a focus for future research.

**Author Contributions:** Conceptualization, Yirong Zhou; Funding acquisition, Jun Li and Luo Chen; Methodology, Yirong Zhou; Project administration, Jun Li and Luo Chen; Resources, Ye Wu and Jiangjiang Wu; Supervision, Jun Li; Writing—original draft, Yirong Zhou; Writing—review & editing, Hao Chen.

**Funding:** National Natural Science Foundation of China: 41871284, 61806211

**Acknowledgments:** This work was supported in part by National Natural Science Foundation of China under Grant No. 41871284 and No. 61806211.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

SLCFF	Station-Level Crowd Flow Forecast
MSVR	Multi-output Support Vector Regression
ARIMA	Auto-Regressive Moving Average
VARIMA	Vector Auto-Regressive Integrated Moving Average
CNNs	Convolutional Neural Networks
LSTM	Long Short Term Memory neural networks

## References

1. Hoang, M.X.; Zheng, Y.; Singh, A.K. FCCF: Forecasting citywide crowd flows based on big data. In Proceedings of the The ACM Sigspatial International Conference, Burlingame, CA, USA, 31 October–3 November 2016; pp. 1–10.
2. Zhang, J.; Zheng, Y.; Qi, D. Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–10 February 2017; pp. 1655–1661.
3. Alexander, L.; Jiang, S.; Murga, M.; González, M.C. Origin–destination trips by purpose and time of day inferred from mobile phone data. *Transp. Res. Part C Emerg. Technol.* **2015**, *58*, 240–250. [[CrossRef](#)]
4. Scholz, R.W.; Lu, Y. Detection of dynamic activity patterns at a collective level from large-volume trajectory data. *Int. J. Geogr. Inf. Sci.* **2014**, *28*, 946–963. [[CrossRef](#)]
5. Kung, K.S.; Greco, K.; Sobolevsky, S.; Ratti, C. Exploring universal patterns in human home-work commuting from mobile phone data. *PLoS ONE* **2014**, *9*, e96180. [[CrossRef](#)] [[PubMed](#)]
6. Zheng, K.; Zheng, Y.; Yuan, N.J.; Shang, S.; Zhou, X. Online discovery of gathering patterns over trajectories. *IEEE Trans. Knowl. Data Eng.* **2014**, *26*, 1974–1988. [[CrossRef](#)]
7. Borchani, H.; Varando, G.; Bielza, C.; Larrañaga, P. A survey on multi-output regression. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2015**, *5*, 216–233. [[CrossRef](#)]
8. Cheng, X.; Ren, L.; Cui, J.; Zhang, Z. Traffic Flow Prediction with Improved SOPIO-SVR Algorithm. In Proceedings of the Monterey Workshop on Challenges and Opportunity with Big Data, Beijing, China, 8–11 October 2016; pp. 184–197.

9. Feng, S.; Hao, C.; Du, C.; Li, J.; Ning, J. A Hierarchical Demand Prediction Method with Station Clustering for Bike Sharing System. In Proceedings of the IEEE Third International Conference on Data Science in Cyberspace, Guangzhou, China, 18–21 June 2018.
10. Hong, W.C.; Dong, Y.; Zheng, F.; Lai, C.Y. Forecasting urban traffic flow by SVR with continuous ACO. *Appl. Math. Model.* **2011**, *35*, 1282–1291. [[CrossRef](#)]
11. Liu, J.; Sun, L.; Li, Q.; Ming, J.; Liu, Y.; Xiong, H. Functional zone based hierarchical demand prediction for bike system expansion. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, 13–17 August 2017; pp. 957–966.
12. Li, Y.; Zheng, Y.; Zhang, H.; Chen, L. Traffic prediction in a bike-sharing system. In Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems, Seattle, WA, USA, 3–6 November 2015; p. 33.
13. Moreira-Matias, L.; Gama, J.; Ferreira, M.; Mendes-Moreira, J.; Damas, L. Predicting taxi—Passenger demand using streaming data. *IEEE Trans. Intell. Transp. Syst.* **2013**, *14*, 1393–1402. [[CrossRef](#)]
14. Zhan, X.; Zheng, Y.; Yi, X.; Ukkusuri, S.V. Citywide traffic volume estimation using trajectory data. *IEEE Trans. Knowl. Data Eng.* **2017**, *27*, 272–285. [[CrossRef](#)]
15. Min, W.; Wynter, L. Real-time road traffic prediction with spatio-temporal correlations. *Transp. Res. Part C Emerg. Technol.* **2011**, *19*, 606–616. [[CrossRef](#)]
16. Asif, M.T.; Dauwels, J.; Goh, C.Y.; Oran, A.; Fathi, E.; Xu, M.; Dhanya, M.M.; Mitrovic, N.; Jaillet, P. Unsupervised learning based performance analysis of n-support vector regression for speed prediction of a large road network. In Proceedings of the 15th International IEEE Conference on the IEEE Intelligent Transportation Systems (ITSC), Anchorage, AK, USA, 16–19 September 2012; pp. 983–988.
17. Chai, D.; Wang, L.; Yang, Q. Bike flow prediction with multi-graph convolutional networks. In Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM), Seattle, WA, USA, 6–9 November 2018; pp. 397–400.
18. Tian, Y.; Pan, L. Predicting short-term traffic flow by long short-term memory recurrent neural network. In Proceedings of the IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity), Chengdu, China, 19–21 December 2015; pp. 153–158.
19. Niepert, M.; Ahmed, M.; Kutzkov, K. Learning convolutional neural networks for graphs. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 17 May 2016, pp. 2014–2023.
20. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; pp. 234–241.
21. Tobler, W.R. A computer movie simulating urban growth in the Detroit region. *Econ. Geogr.* **1970**, *46*, 234–240. [[CrossRef](#)]
22. Defferrard, M.; Bresson, X.; Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; pp. 3844–3852.
23. Berkhin, P. A survey of clustering data mining techniques. In *Grouping Multidimensional Data*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 25–71.
24. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
25. Chen, T.; Guestrin, C. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd ACM Sigkdd International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794.

