A Hybrid Integrated Deep Learning Model for Citywide Spatio-temporal Flow Volume Prediction

Journal:	*International Journal of Geographical Information Science*
Manuscript ID	IJGIS-2018-0218.R4
Manuscript Type:	Special Issue Paper
Keywords:	spatio-temporal flow volume, prediction, deep learning, LSTM, ResNet

A Hybrid Integrated Deep Learning Model for Citywide Spatio-temporal Flow Volume Prediction

Abstract

Recently, the spatio-temporal residual network (ST-ResNet) has leveraged the power of deep learning (DL) to predict citywide spatio-temporal flow volume. However, this model, neglects the dynamic dependency of the input series in the temporal dimension, which affects the captured spatio-temporal features. The present study introduces the long short-term memory (LSTM) neural network into ST-ResNet, to form a hybrid integrated DL model for citywide spatio-temporal flow volume prediction (called HIDLST). The new model is capable of dynamically learning the temporal dependency via the feedback connection of the LSTM, improving the accuracy of the spatio-temporal features. We test the HIDLST model by predicting the citywide taxi flow volume in Beijing, China. We tune the hyperparameters of the HIDLST model to optimize the prediction accuracy. Comparative experiments indicate that the proposed model consistently outperforms ST-ResNet and several other typical DL-based models with regard to prediction accuracy. Additionally, we also discuss the distribution of prediction errors and the contributions of different spatio-temporal patterns.

Keywords: spatio-temporal flow volume; prediction; deep learning; LSTM; ResNet

1 Introduction

1.1 Background and purpose

Human mobility is a central theme in human geography and urban analytics. People

constantly interact with the urban space through various spatio-temporal activities, such as taking buses, driving, and walking (Zhu et al. 2018). Most of the trajectories generated by these activities can be recorded (Shaw et al. 2016, Zhang and Weghe 2018), resulting in abundant datasets that provide spatial and temporal knowledge (Guo et al. 2012, Gao et al. 2013, Gong et al. 2016, Shen and Cheng 2016). Spatio-temporal flow volume data is generated via statistical analyses of these trajectories (Zhu and Guo 2014). As shown in Figure 1(a), for the spatial unit *i*, objects 1 and 2 are two inflow objects, and object 3 is an outflow object. Hence, the inflow volume is two, and the outflow volume is one. Given a city divided into a grid with *M* rows and *N* columns, a time interval *t*, and a trajectory dataset, the inflow and outflow volume of each grid cell can be calculated (Zhang et al. 2016). The three-dimensional tensor $X_t^{2 \times M \times N}$ represents the citywide flow volume, where "2" corresponds to inflow and outflow, as shown in Figure 1(b) (Zhang et al. 2018). Herein, the term "spatio-temporal flow volume" refers to the separate inflow and outflow volumes. Figure 1(c) shows an instance generated from taxi trajectories in Beijing, China with a 32 × 32 grid partition and a 30-min time interval.

Figure 1. Citywide spatio-temporal flow volume

The citywide spatio-temporal flow volume quantitatively reflects the distribution of moving objects over space and time (Zheng et al. 2014). Predicting the future volumes of traffic or crowds on a citywide scale helps the urban manager to prevent traffic congestion and stampedes (Silva et al. 2015, Chen et al. 2015, Hoang et al. 2016). The objective of this study is to construct a prediction model that can

learn features from historical observations and accuratly predict the citywide spatio-temporal flow volume according to these features.

In the machine learning (ML) domain, a feature is an individual measurable property of the phenomenon being observed (Bishop 2006). The key to achieving a high-accuracy prediction system is capturing features as accurately as possible (LeCun et al. 2015). For citywide spatio-temporal flow volume prediction, the most important feature is the spatio-temporal dependency (Zhang et al. 2018, Chen et al. 2018). The spatial dependency refers to the interactions between the inflow and outflow of near and distant neighbors. For example, the traffic flow on a road that crosses another affects the traffic on the second road. Numerous individuals drive to offices over various distances, generating distant dependency. The temporal dependency may arise from different patterns. For example, the traffic flow at 7:00 am may have a strong correlation with that at 6:00 am. It may also be very similar to the flow at 7:00 am on the previous day or that at 7:00 am on the same day of the previous week, because human activities have daily and weekly periodicities. The biggest challenge for spaito-temporal flow volume prediction is capturing spatio-temporal dependency.

Recently, the convolution neural network (CNN), a well-known deep learning (DL) method, has been employed to automatically capture the citywide spatio-temporal dependency. While the CNN is effective for modeling spatial dependency, it is not suitable for capturing dynamic temporal dependency, which is significant in spatio-temporal prediction (Cheng and Wang 2009). In contrast, the long short-term

memory (LSTM) neural network is a dynamic deep neural network for capturing temporal dependency. The purpose of this study is to introduce LSTM into the CNN to form a hybrid integrated DL model for citywide spatio-temporal flow volume prediction (called HIDLST). The proposed model can capture both the dynamic temporal dependency and the spatial dependency automatically and consistently, improving the prediction accuracy for flow volumes.

1.2 Related work and existing problem

Most existing spatio-temporal prediction models derive from statistical and ML methods, including the space-time autoregressive integrated moving average (Wang et al. 2010, Cheng et al. 2014), the space-time support vector regression (Wang et al. 2007), and the space-time artificial neural network (ANN) models (Wang et al. 2016, Chen et al. 2018). However, these conventional models are incapable of accepting raw input datasets. When constructing an ML model including the aforementioned ones, the feature extraction requires careful engineering and considerable domain expertise for transforming raw data into a proper internal representation for spatio-temporal dependency detection. This procedure is called "feature engineering" (LeCun et al. 2015). In the Big Data era, the feature engineering is particularly challenging.

DL addresses this challenge (Hinton and Salakhutdinov 2006). A typical DL model can accept raw input data and automatically discover the required features. This is called "end-to-end" learning and enormously simplifies the feature

engineering process. The LSTM (Hochreiter and Schmidhuber 1997) is a special type of deep recurrent neural network (RNN) that dynamically feeds the output of the previous step back into the input layer of the current step, which is called a dynamic feedback connection. The output depends on both the current input and the previous features. The feedback connection makes LSTM particularly suitable for capturing the dynamic temporal dependency occurring in a time series (Cheng and Wang 2009). Hence, researchers have proposed several LSTM-based models for obtaining better accuracy than traditional methods for predicting traffic data (Ma et al. 2015, Wang et al. 2017). However, LSTM cannot capture spatial dependency, which is extremely important for spatio-temporal flow volume prediction.

To address this issue, researchers have introduced the CNN into spatio-temporal flow volume prediction. A CNN unit is a neural network that connects to the local patches in the feature maps of the previous layer through a set of weights called convolution kernels (LeCun et al. 2015). Stacking multiple CNN layers allows distant spatial dependency to be captured. Zhang et al. (2016) used a grid map to represent the citywide spatio-temporal flow volume. They designed a multi-layer CNN structure (called DeepST) to receive historical observations from hourly, daily, and weekly patterns for learning the spatial and temporal dependencies simultaneously. DeepST can achieve “end-to-end” predictions for an entire city. Zhang et al. (2018) integrated the deep residual network (He et al. 2016a) into DeepST, forming the deep spatio-temporal residual network (ST-ResNet). The residual network significantly increases the depth of the neural network model (He et al. 2016a). A deeper CNN

structure forms a broader spatial receptive field to capture the spatial dependency from distant regions. ST-ResNet exhibited an accuracy higher than that of DeepST (by 7.09%) for predicting the citywide taxi flow volume in Beijing, China (Zhang et al. 2018).

The CNN is a typical static neural network without a dynamic feedback connection, and it is difficult to capture the dynamic dependency occurring in time series. Therefore, the CNN-based model ST-ResNet has limited prediction accuracy for the spatio-temporal flow volume. Several studies have confirmed the advantage of capturing temporal dependency dynamically in spatio-temporal prediction (Cheng and Wang 2008, Cheng et al. 2008, Cheng and Wang 2009).

1.3 Proposed solution

LSTM is an RNN that has a feedback connection and is capable of capturing the dynamic temporal dependency in a time series. The residual CNN (ResNet) can capture the spatial dependency well. We combine these two models to capture the spatio-temporal dependency more accurately. First, the LSTM structure captures the temporal dependency, and then the deep ResNet captures the spatio-temporal dependency from the LSTM outputs. The temporal dependency captured by the LSTM structure contains the dynamic dependency occurring in the time series, which can characterize the data more accurately than CNN-based models. Finally, the spatio-temporal dependency captured by ResNet is suitable for achieving better prediction accuracy.

This study proposes a hybrid integrated DL model for citywide spatio-temporal flow volume prediction (HIDLST) by integrating LSTM and ResNet. Compared with the existing models, the proposed model can automatically and accurately capture both the spatial and dynamic temporal dependencies in spatio-temporal flow volume data.

The remainder of this paper is organized as follows. Section 2 defines the problem of citywide spatio-temporal flow volume prediction and details the structure of the proposed HIDLST model. Section 3 introduces a case study involving prediction of the citywide taxi flow volume in Beijing, China. Section 4 discusses the distribution of prediction errors and the contributions from different spatio-temporal patterns. Finally, Section 5 presents the conclusions and directions for future work.

2 Methodology

2.1 Problem definition

For the prediction of the citywide spatio-temporal flow volume, the most important feature is the spatio-temporal dependency. Other relevant factors include the weather conditions, temperature, and holidays, as these factors influence the travel paths, time, and types of human activities (Hoang et al. 2016, Ke et al. 2017).

Equation (1) defines the citywide spatio-temporal flow volume prediction problem.

$$X_T = F_{predict}(X_{ST}, X_E, W) \quad (1)$$

Here, X_T represents the prediction target (spatio-temporal flow volume data at the T^{th}

time interval), $F_{predict}$ represents the prediction model to be constructed, X_{ST} represents the set of historical spatio-temporal flow volume observations, X_E represents the external factors, and W represents the parameters to be learned. Figure 2 shows an example procedure of $F_{predict}$, which is used to learn the spatio-temporal dependency and the external impacts (W) from historical sets X_{ST} and X_E and make predictions regarding future flow volumes.

Figure 2. Problem definition of citywide spatio-temporal flow volume prediction

2.2 Hybrid integrated DL model for citywide spatio-temporal flow volume prediction

Figure 3 shows the general framework of the proposed HIDLST. The spatio-temporal dependency arises from different patterns (Ma et al. 2014, Wang et al. 2017, Zhang et al. 2018). We divide the historical spatio-temporal flow volume data into hourly, daily, and weekly patterns and construct three sub-models with the same structure to capture the features.

First, an LSTM structure receives the raw spatio-temporal flow volume grids to capture the dynamic temporal dependency occurring in the time series, forming candidate feature maps. A multi-layer ResNet model simultaneously captures the spatio-temporal dependency by performing convolutions on the candidate feature maps, resulting in three spatio-temporal feature maps (ST-maps). Next, we merge the ST-maps into a two-channel raster map (final ST-map). The final ST-map combines the external factors to obtain prediction results. Finally, the model calculates the loss

and optimizes the parameters via back-propagation. The following sections detail the main modules of HIDLST.

Figure 3. Framework of HIDLST

2.2.1 Input datasets

The input datasets consist of spatio-temporal flow volume data from hourly, daily, and weekly patterns. Hourly inputs refer to historical observations that are close to the target time. Daily and weekly inputs refer to historical observations at the same time point as the prediction target but with daily or weekly periodicity. For example, assume that the prediction target is the spatio-temporal flow volume at 9 am on Thursday and the lengths of the hourly, daily, and weekly patterns are three, one, and one, respectively. The hourly inputs are the historical observations at 8:30 am, 8:00 am, 7:30 am, and so on. The daily inputs are the observations at 9 am on the days prior, and the weekly inputs are the observations at 9 am on the previous Thursdays. To eliminate the fluctuation in adjacent time intervals, we add a time buffer to the daily and weekly patterns (Wu and Tan 2016). Let t represent the target time. The number of time intervals in 1 d is m, and the radius of the time buffer is b. The spatio-temporal flow volume data at the i^{th} time interval are represented by X_i, which is a three-dimensional tensor, as mentioned in Section 1.1. The three input datasets X_H, X_D, and X_W are four-dimensional tensors whose dimensions are $h \times 2 \times M \times N$, $d \times 2 \times M \times N$ and $w \times 2 \times M \times N$, respectively, as indicated by Equation (2).

$$X_H^{h\times 2\times M\times N} = (X_{t-h},...,X_{t-i},...,X_{t-2},X_{t-1}),\{i\in Z\,|\,1\le i\le h\}$$

$$X_D^{d\times 2\times M\times N} = (X_{t-j\cdot m-b}\ldots X_{t-j\cdot m-1},X_{t-j\cdot m},X_{t-j\cdot m+1}\ldots X_{t-j\cdot m+b}),\{j\in Z|1\le j\le d$$

$$X_D^{w\times 2\times M\times N} = (X_{t-k\cdot 7\cdot m-b}\ldots X_{t-k\cdot 7\cdot m-1},X_{t-k\cdot 7\cdot m},X_{t-k\cdot 7\cdot m+1}\ldots X_{t-k\cdot 7\cdot m+b}),\ \{k\in Z\,|\,1\le k\le w\} \quad (2)$$

Here, h, d, and w refer to the numbers of time intervals corresponding to hourly, daily, and weekly patterns, respectively, and X_i has the dimensions $2 \times M \times N$. Figure 3(a) shows an instance with $h = 3$, $d = 1$, $w = 1$, and $b = 1$.

2.2.2 Integrally capturing spatio-temporal dependency

The core procedure of HIDLST includes two components: capturing the temporal dependency using the LSTM structure(Figure 3-(b)) and capturing the spatio-temporal dependency using ResNet (Figure 1 (c)). We take the hourly pattern as an example to detail the procedure.

Figure 4. Integrally capturing the spatio-temporal dependency

(1) Capturing temporal dependency using LSTM

An LSTM module receives the raw spatio-temporal flow volume grids. Figure 4(a) unfolds the feedback connection of the LSTM and details its structures. The memory cell of the i^{th} time step (C_i) is accessed, written, and cleared by the input gate (i_i), forget gate (f_i), and output gate (o_i), respectively (Hochreiter and Schmidhuber 1997, Graves et al. 2013). Taking the second time step input X_2 as an example, if i_2 is activated, the temporal dependency of the current step is accumulated to C_2. Similarly, f_2 determines whether C_2 forgets the dependency of the previous step stored

in C_1, and o_2 determines whether the dependency in C_2 is propagated to the current output Y_2, which will be the input for both the next time step and the next LSTM layer. For a time series, the step-by-step feedback captures the dynamic temporal dependency between all time steps and the target. The output of the last step (t-1) forms the candidate feature map $C^{O\times M\times N}$, where O represents the number of hidden neurons in the LSTM layer, and $M \times N$ represents the grid map. These candidate feature maps contain the temporal dependency and spatial information used to capture the spatio-temporal dependency. The transformation is summarized by Equation (3).

$$C^{O \times M \times N} = F_{LSTM}(X_H, W_{LSTM}) \quad (3)$$

Here, F_{LSTM} represents the transformation of the LSTM model, and W_{LSTM} represents all the parameters learned during the procedure.

(2) Capturing spatio-temporal dependency using ResNet

The multi-layer ResNet module accepts the outputs of the LSTM to capture the spatio-temporal dependency (Figure 4b). Each ResNet unit comprises a stack of two "Rectified Linear Unit (ReLU) + CNN" layers with a shortcut connection linking the input and output of the second CNN layer (He et al. 2016b, Zhang et al. 2018) (Figure 4c). As $C^{O\times M\times N}$ contains the temporal dependency and spatial information, the following CNN units execute convolution to capture the spatio-temporal dependency. The ReLU (Nair and Hinton 2010) performs activation to model non-linear features.

The shortcut connection of the ResNet unit requires that the input grids and the output grids have the same shape (He et al. 2016b). The shape of the output grids is determined by the number of convolution kernels (denoted as K) in the ResNet unit.

Thus, a convolution layer (denoted as CNN_1), converts the output of LSTM $C^{O\times M\times N}$ to the shape of the ResNet output (denoted as $C^{K\times M\times N}$) according to Equation (4).

$$C^{K\times M\times N} = F_{conv1}(C^{O\times M\times N}, K, W_1) \quad (4)$$

Here, F_{conv1} represents the convolution operation, and W_1 is the trainable parameter.

Assuming that the convolution kernel size is r, after L ResNet units, Equation (5) calculates the spatial radius of the accumulated receptive field S. As the length of the raw input series is h, for each grid cell, ResNet captures the features that are integrally spatio-temporally correlated with its S-1 spatial orders and h temporal orders, outputting the feature maps $ST^{K\times M\times N}$ (Figure 4(d)). The procedure is described by Equation (6).

$$S = (r-1)\cdot L + 1 \quad (5)$$

$$ST^{K\times M\times N} = F_{resnet}(C^{K\times M\times N}, W_{resnet}) \quad (6)$$

Here, F_{resnet} represents the ResNet transformation, and W_{resnet} represents all the parameters learned during the procedure. Finally, we convert $ST^{K\times M\times N}$ to a two-channel grid $STFM_H$ (hourly ST-map) by CNN_2 to calculate the loss directly, as indicated by Equation (7).

$$STFM_H^{2\times M\times N} = F_{conv2}(ST^{K\times M\times N}, 2, W_2) \quad (7)$$

Here, F_{conv2} represents the convolution operation, and W_2 is the trainable parameter.

2.2.3 Feature fusion

The feature fusion procedure includes two steps: spatio-temporal feature fusion and external factor fusion. We adopt the same fusion method used in ST-ResNet (Zhang et

al. 2018). $STFM_H$, $STFM_D$, and $STFM_W$ represent the spatio-temporal feature maps of the hourly, daily, and weekly patterns, respectively. As confirmed by Zhang et al. (2018), the influence varies across temporal patterns and regions. The parametric-matrix-based method (Zhang et al. 2018) fuses the three maps into a final spatio-temporal feature map (Equation (8)). $STFM_A$ represents the fusion result; W_H, W_D, and W_W represent three parameter matrices with the same shapes as $STFM_H$, $STFM_D$, and $STFM_W$, respectively; and "∘" represents the Hadamard product. The final spatio-temporal feature map fuses the external factors. The external factors in this study include weather, temperature, wind speed, and holidays. A two-layer ANN module embeds all the external factors (Zhang et al. 2018) in a two-channel grid $E^{2\times M\times N}$. Finally, the model fuses $STFM_A^{2\times M\times N}$ and $E^{2\times M\times N}$ directly. The *tanh* function activates the fusion result to obtain the final prediction values, which are represented by $X_t^{2\times M\times N}$ in Equation (9).

$$STFM_A^{2\times M\times N} = STFM_H \circ W_H + STFM_D \circ W_D + STFM_W \circ W_W \quad (8)$$

$$X_t^{2\times M\times N} = \tanh\left(STFM_A^{2\times M\times N} + E^{2\times M\times N}\right) \quad (9)$$

2.2.4 Model training

The mean-squared error (MSE) is the loss function. In Equation (10), y_i represents the ground truth, y_i' represents the prediction value, and N represents the number of values to be predicted. We divide all the samples into three sub-datasets: a training set, a validation set, and a testing set. We feed the training set into the model in batches. For each batch, the model calculates the loss after forward propagation.

Then, an optimizer updates all the training parameters via back-propagation. An Adam (Kingma and Ba 2014) optimizer generates adaptive learning rates for different parameters. By minimizing the loss, all the trainable parameters are trained.

$$\text{Loss} = \text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - y_i')^2 \quad (10)$$

3 Case Study

3.1 Experiment data and environment

We validate the HIDLST with a spatio-temporal flow volume dataset generated from taxi GPS trajectories in Beijing, China from November 1st 2015 to April 9th 2016 (Zhang et al. 2018). The study area is a 32-km^2 square region located in the main districts of Beijing (Figure 5(a)). We divide the area into a 32 × 32 grid, with a cell size of 1 km^2. The time interval is 30 min. Figure 5(b) shows the main roads in the study area.

Figure 5. Study area

A week is the smallest unit that contains both workdays and a weekend. We selected the data from the last week (April 3rd 2016 to April 9th 2016) as the testing set. Similar to most supervised learning systems (LeCun et al. 2015), to tune the hyperparameters, we divide the remaining data into a training set and a validation set, with a proportion of 9:1. To increase the convergence speed, we normalize the values of the spatio-temporal flow volume into the range [-1, 1] for the training (Ioffe and Szegedy 2015) and transforms all prediction values into the normal values for

evaluation. The external factors are weather conditions, temperature, wind speed, and holidays. The pre-processing procedure is identical to that used in a previous study (Zhang et al. 2018). We employ the root-mean-square error (RMSE), i.e., the square root of the MSE given by Equation (10), as the evaluation metric. We adopt the early-stopping strategy (Caruana et al. 2001) to avoid overfitting. When the RMSE of the validation set does not decay for five iterations, the training procedure will be stopped.

We code the model in Python 3.5, using Keras (Chollet 2015) and TensorFlow (Abadi et al. 2016) as the DL packages. The LSTM module is the Keras layer called ConvLSTM (Shi et al. 2015), which is a model integrating convolution operations into an LSTM unit. A ConvLSTM layer can process two-dimensional grid data (such as an image), similar to a CNN layer. We set the kernel size of the ConvLSTM unit to one; thus, such that the LSTM module can directly accept and output spatio-temporal flow volume girds. We perform all experiments using a graphics unit (GPU) platform (NVIDIA GeForce GTX 1080 with 8GB of GPU memory).

3.2 Tuning parameters

Parameter tuning is essential for identifying the optimal parameters for DL-based models (Ma et al. 2017, Ke et al. 2017, Zhang et al. 2018). We tune the hyperparameters to obtain the optimal prediction results of HIDLST. There are four main hyperparameters: the number of hidden neuron units, number of hidden layers, lengths of different input patterns, and convolution kernel size.

3.2.1 Number of hidden neuron units

FN_{LSTM} and FN_{ResNet} represent the numbers of hidden units in LSTM and ResNet, respectively. To test the LSTM, FN_{ResNet} is fixed at 64, and FN_{LSTM} varies between four, eight, 16, and 32. Correspondingly, for testing ResNet, FN_{LSTM} is fixed at 32, and FN_{ResNet} varies between four, eight, 16, 32, and 64. The LSTM has two hidden layers, and the number of ResNet units increases from one to 13. We record the minimum RMSE. Figure 6 shows the results. The model exhibits the best prediction accuracy (RMSE 14.6) when FN_{LSTM} is 32 and FN_{ResNet} is 64.

(a) (b)

Figure 6. Experimental results for different hidden neuron units

3.3.2 Number of hidden layers

We perform two experiments to determine the optimal depth (number of hidden layers) for HIDLST. We investigate the depths of the LSTM and ResNet separately, which are denoted as D_{LSTM} and D_{ResNet}, respectively. The other parameters remain the same. First, with $D_{LSTM} = 2$, D_{ResNet} varies from one to 13 in step two. The RMSE initially decreases with the increase in D_{ResNet}, reaching its minimum value when D_{ResNet} is five. Subsequently, the RMSE increases (Figure 7(a)). Adding layers introduces training difficulty and noise at the fringe of the city. For the LSTM, D_{ResNet} is 5, and D_{LSTM} varies from one to four in step one. The overall trend is similar to that for ResNet, and the minimum RMSE is obtained with two LSTM layers (Figure 7(b)).

Thus, the optimal depths of HIDLST are two LSTM layers and five ResNet units.

(a) (b)

Figure 7. Experimental results for different model depths

3.3.3 Lengths of different input patterns

We define L_H, L_D, and L_W as the lengths of the hourly, daily, and weekly patterns, respectively. The other parameters remain the same. First, with $L_D = 1$ and $L_W = 1$, L_H varies from zero to eight in step two. Figure 8(a) presents the results. The RMSE is the largest when L_H is zero (when the model omits the hourly pattern). Thus, the hourly pattern is crucial. The RMSE is minimized at $L_H = 4$. For $L_H = 4$, $L_W = 1$, and $L_D = \{0, 1, 2, 4, 6\}$ (Figure 8(b)), the trend is similar to that for the hourly pattern. The RMSE is minimized at $L_D = 1$. The RMSE increases when L_D is larger than one. For example, the situation at 9:00 am on Sunday is not closely related to that at 9:00 am on Monday ($L_D = 6$) but is closely related to that at the same time on Saturday ($L_D = 1$). Lastly, regarding the weekly patterns, Figure 8(c) shows that the optimal length is $L_W = 1$. The best lengths for the hourly, daily, and weekly input patterns are four, one, and one, respectively.

(a) (b) (c)

Figure 8. Experimental results for different input lengths

3.3.4 Convolution kernel size

To identify the optimal kernel size, we change the kernel sizes to 3 × 3, 5 × 5, and 7 × 7. The depth of ResNet increases gradually, as before, and the minimum RMSE is recorded. The other settings are the same. As indicated by Table 1, the best convolution kernel size for the model is 3 × 3. Although a larger kernel covers a larger spatial region, the number of trainable parameters increases with the kernel size. With the same receptive field, stacking small multi-layer kernels allow the detection of more complex non-linear features compared with the case of a single-layer larger kernel. Many image-processing applications employ the 3 × 3 kernel (Simonyan and Zisserman 2014, He et al. 2016a). For HIDLST, 3 × 3 is the optimal kernel size.

Table 1. Experimental results for different convolution kernel sizes

No.	Kernel Size	RMSE
1	3 × 3	14.6
2	5 × 5	14.91
3	7 × 7	16.32

Table 2 presents the optimal hyperparameters for HIDLST. The model obtains the minimum RMSE (14.6) with the optimal setting.

Table 2. Optimal hyperparameters for HIDLST

Number of hidden neuron units	Number of hidden layers	Lengths of input patterns	Convolution kernel size
$FN_{LSTM} = 32$	$D_{LSTM} = 2$	$L_H = 4$, $L_D = 1$, L_W	$K = 3$

$FN_{ResNet} = 64$	$D_{ResNet} = 5$	$= 1$

3.3 Comparative experiments

We compare HIDLST with five other DL-based models (Table 3). All these models take the hourly, daily, and weekly data as three separate inputs and fuse the feature maps of the three patterns later. The external factors are fused via the same method. The Adam optimizer is used for all the models. To avoid overfitting, if the RMSE of the validation dataset does not decrease after five loops, the training procedure will stop.

Table 3. Descriptions of the different models

No.	Model	Description
1	LSTM	It consists of multiple stacked LSTM layers, without ResNet.
2	ConvLSTM	A stacked multi-layer ConvLSTM model (Xingjian et al. 2015). The ConvLSTM unit can perform convolution operations in the LSTM memory block.
3	ST-ResNet	It consists of multiple ResNet layers, without an LSTM filter. It is a state-of-the-art model (Zhang et al. 2018).
4	ST-ResNet-TB	It has the same main structure as ST-ResNet, except for the input data. It has time buffers for the daily and weekly inputs, similar to the HIDLST model.
5	Hybrid-LR	It is a hybrid model including LSTM and ResNet modules. However, the inputs are fed into a ResNet model and an LSTM model separately, and the outputs are merged via a parametric-

		matrix-based method (similar to Equation (8)). It is similar to the model proposed by Wu and Tan (2016), except that the CNN is replaced with ResNet.
6	HIDLST	The proposed hybrid spatio-temporally integrated DL model.

Similar to the case of the HIDLST model, we tune the hyperparameters for the other five baseline models and record the minimum RMSEs. Table 4 presents the optimal hyperparameters settings, minimum RMSE, and convergence time and iterations for each model. FN_{LSTM} represents the number of hidden neurons of one LSTM layer. FN_{ResNet} represents the number of hidden neurons of one ResNet unit. D_{LSTM}, $D_{ConvLSTM}$, and D_{ResNet} represent the numbers of hidden layers in LSTM, ConvLSTM, and ResNet, respectively. L_H, L_D, and L_W represent the lengths of the hourly, daily, and weekly patterns, respectively. K represents the convolution kernel size. For the ST-ResNet model, to keep the inputs the same as those of the original model, the time buffer (denoted b) is zero in the daily and weekly inputs. For the other five models, b was one.

Table 4. RMSE values of the different models

No.	Model	Optimal hyperparameters	RMSE	Iterations/ Time
1	LSTM	$FN_{LSTM} = 64$ $D_{LSTM} = 3$ $L_H = 4, L_D = 1, L_W = 1$	16.59	118/5005 s
2	ConvLSTM	$FN_{ConvLSTM} = 64$ $D_{ConvLSTM} = 4$	15.76	201/10940 s

		$L_H = 4, L_D = 1, L_W = 1$ $K = 3$		
3	ST-ResNet	$FN_{ResNet} = 64$ $D_{ResNet} = 10$ $L_H = 3, L_D = 1, L_W = 1$ $K = 3$	15.81	101/1798 s
4	ST-ResNet-TB	$FN_{ResNet} = 64$ $D_{ResNet} = 10$ $L_H = 3, L_D = 1, L_W = 1$ $K = 3$	15.32	87/1602 s
5	Hybrid-LR	$FN_{LSTM} = 64, FN_{ResNet} = 64$ $D_{LSTM} = 2, D_{ResNet} = 8$ $L_H = 4, L_D = 1, L_W = 1$ $K = 3$	15.37	101/2651 s
6	HIDLST	$FN_{LSTM} = 32, FN_{ResNet} = 64$ $D_{LSTM} = 2, D_{ResNet} = 5$ $L_H = 4, L_D = 1, L_W = 1$ $K = 3$	**14.60**	120/2903 s

As shown in Table 4, HIDLST has the smallest RMSE among the models. Compared with the LSTM model, HIDLST exhibits a reduction of 11.99% in the RMSE. For the citywide spatio-temporal flow volume, it is difficult to achieve accurate predictions by only capturing the temporal dependency.

The RMSE of the classical ST-ResNet model is 15.81. ST-ResNet -TB reduces the RMSE to 15.32 by adding time buffers; thus, the time buffers are

effective. Compared with these two ST-ResNet-based models, the HIDLST exhibits reductions of 7.65% and 4.70%, respectively, in the RMSE. We employ the Independent-Sample T Test (Heeren and D'Agostino 1987) to determine whether there is a statistically significant difference between the errors of ST-ResNet-TB and HIDLST. The null hypothesis is that there is no significant difference. Table 5 shows that the p-value is 2.53E-71 (less than 0.05); thus, the null hypothesis is rejected. Therefore, the difference between the errors of these two models is significant. To investigate this further, for HIDLST and ST-ResNet-TB, we calculate the RMSE of each time interval based on 2,048 ($2 \times 32 \times 32$) values, and then plot the distributions covering 336 predicted time intervals (Figure 9). HIDLST outperforms ST-ResNet-TB in nearly 70% of the time intervals (the exact number is 234), with a smaller RMSE. We select three representative cells with a high mean ground truth (MGT) labeled as cells 1, 2, and 3 in Figures 10(a) and (b) to plot the ground truths, predictions of HIDLST, and predictions of ST-ResNet-TB, as shown in Figure (c)-(h). The MGT is defined by Equation (11), where y_i represents the ground truth, and T represents the number of time intervals in the sub-dataset. Generally, HIDLST fit the ground truths better, particularly at the points marked by circles.

$$\mathrm{MGT} = \frac{1}{T}\sum_{i=1}^{T} y_i \tag{11}$$

Therefore, compared with the ST-ResNet model, the new proposed model significantly improved the capturing of spatio-temporal features by dynamically capturing the temporal dependency with LSTM. Thus, it achieves a higher prediction accuracy.

Table 5. Independent-Samples T test results for ST-ResNet-TB and HIDLST

t-value	p-value
17.86	2.53E-71

Figure 9. RMSEs of HIDLST and ST-ResNet-TB for all the predicted time intervals

Figure 10. Ground truths, predictions of HIDLST, and predictions of ST-ResNet-TB

Moreover, the HIDLST model exhibits an RMSE reduction of 5.01% compared with the Hybrid-LR model. By separately capturing the temporal and spatial dependencies and recombining them, the advantages of LSTM and ResNet cannot be fully exploited, because there is no direct interaction between the two models. Conversely, the integrated method employer by HIDLST fully exploits the advantages of the two models.

Lastly, the HIDLST model exhibits a distinct advantage over the ConvLSTM model. We obtain the minimum RMSE (15.76) with four stacked ConvLSTM layers. Although ConvLSTM performs convolution and LSTM operations simultaneously, it is difficult to train the model in a deep mode, which limits the range of the spatially receptive field for capturing the spatio-temporal dependency. The ResNet in the HIDLST model solves this problem.

ConvLSTM spends the longest time and requires the largest number of steps to achieve convergence. LSTM spends the second-longest time. The two ST-ResNet-based models spend less time. This is mainly because the convolution operation can be fully parallelized during execution, but the LSTM operation cannot (Lei et al. 2017). Thus, the computation time of HIDLST (2903 s) is longer than those of the ST-ResNet-based models; however, it is reasonable.

In summary, by integrating LSTM and ResNet, the HIDLST model can capture the spatio-temporal dependency more accurately than several existing DL-based spatio-temporal flow volume prediction models, with a reasonable computation time.

4 Discussion

4.1 Distribution of errors

To analyze the spatio-temporal distribution of prediction errors, we divide the test dataset into four segments: workday non-sleeping hours (07:00–24:00), workday sleeping hours (00:00–07:00), weekend (holiday) non-sleeping hours (07:00–24:00), and weekend (holiday) sleeping hours (0:00–7:00). The workdays are from Tuesday to Friday. Monday (April 3rd 2016) is a holiday, and Saturday and Sunday are the weekend. For each sub-dataset, we calculate the MGT and RMSE for each grid cell.

Figure 11. Distributions of the MGT and RMSE

Figure 11 shows the results. From a temporal perspective, although the MGTs for the sleeping hours are significantly lower than those for the non-sleeping hours, the corresponding RMSEs do not exhibit the same pattern. Figure 12(a) displays the relationship between the RMSE and MGT. For most cells, the errors for the sleeping hours are larger than those for the non-sleeping hours. To investigate the reason for this, for each grid cell, we calculate the variances of the MGT for the sleeping and non-sleeping hours. We examine the relationship between the RMSE and the variances of the MGT, as well as the relationship between the variances of MGT and the MGT. The RMSE exhibits a strong positive correlation with the variance of the MGT (Figure 12(b)). As shown in Figure 12(c), the variances of the MGT for the sleeping hours are larger than those for the non-sleeping hours. Consequently, the lower MGTs for the sleeping hours generate larger RMSEs. The larger MGT variances correspond to more variable human activities occurring during sleeping hours, which are more difficult to characterize using the prediction model.

Figure 12. Relationships between the MGT and RMSE, between the RMSE and the variance of the MGT, and between the variance of the MGT and the MGT

For a spatial perspective, the errors along the airport expressway (cells located in the upper-right light rectangle in Figure 11(b)) are larger than those for other areas, particularly for the sleeping hours. This is possibly due to the weak connectivity of the airport expressway with the surrounding cells. Because the airport expressway connects the central urban area to the airport and has several fixed entrances and exits,

the interactions between the cells on this road and the adjacent cells off of this road only occur at fixed intersections; thus, the connection is sparse. Most of the cells surrounding the airport expressway are urban suburbs with a very small flow volume, particularly during the sleeping hours. All the surrounding cells are included in the convolution, introducing noise. Consequently, the airport expressway regions exhibit relatively large RMSE values (Figures 11(b) and (d)).

The HIDLST model does not perform well with activities that are variable and regions where the spatial connectivity is weak.

4.2 Contributions from three spatio-temporal patterns

In the feature fusion stage (Section 2.2.3), we create three weight maps (W_H, W_D, and W_W) to merge the spatio-temporal features learned from the hourly, daily, and weekly patterns (Equation (8)). To analyze the contributions of the different patterns, we normalize the weights of the inflow and outflow to [0, 1] for each cell and plot their distributions, as shown in Figure 13. Figures 13(a)-(f) exhibit that the weights of each temporal pattern present varying degrees of spatial heterogeneity. While the hourly weights vary relatively little among different grid cells, the daily and weekly weights vary significantly. The heterogeneity does not exhibit obvious spatial patterns. However, for the hourly pattern of the inflow volume, the cells whose weights are <0.25 account for 17.68% of the total number of cells, and for the daily and weekly patterns, the percentages are 33.89% and 30.47%, respectively (Figure 13(g)). The weights of >70% of the cells for the hourly pattern are between 0.25 and 0.50 (far

higher than those for the other two patterns, as shown in Figure 13(g)), which is consistent with the small spatial variations in Figure 13(a). The outflow exhibits similar patterns to the inflow (Figure 13(h)). Thus, the contribution of the hourly pattern is relatively important for most cells.

Figure 13. Weight matrix distributions for the hourly, daily, and weekly patterns

5 Conclusions and future work

This study proposes a new method, i.e., HIDLST, to predict the citywide spatio-temporal flow volume by integrating two DL methods:LSTM and ResNet. A major advantage of this hybrid model over the state-of-the-art ST-ResNet model (Zhang et al. 2018) is its capability to dynamically capture the temporal dependency in spatio-temporal flow volume series. We test the proposed model via a case study involving prediction of the citywide taxi flow in Beijing, China for a week. The experimental results indicate that the HIDLST model significantly outperforms several existing DL-based models (LSTM, ConvLSTM, Hybrid-LR, and ST-ResNet) and has a reasonable computation time. A detailed comparison between HIDLST and ST-ResNet-TB reveals that HIDLST has higher performance in nearly 70% of the time intervals. For key regions with high flow volumes, HIDLST fits the ground truths better. In summary, HIDLST can automatically and accurately capture both the spatial and dynamic temporal dependencies in citywide spatio-temporal flow volume data. Accurate prediction of the citywide spatio-temporal flow volume can help urban

managers make effective plans to deal with various situations. The prediction results can also provide references for people's travel time and travel choice.

The hourly pattern exhibit relatively important contributions to more areas than the daily and weekly patterns. The weights of each temporal pattern exhibit varying degrees of spatial heterogeneity. However, the variations do not exhibit obvious spatial patterns.

This study has several limitations. First, the prediction errors of HIDLST are relatively large for sleeping hours and areas with sparse spatial connections (e.g., the airport expressway). Second, in this study, the size of the grid cell is 1 km^2, and the time interval is 30 min. The relationship between the spatio-temporal resolutions and the prediction errors is not fully explored. In the future, we will address these limitations. We will also investigate the applicability of HIDLST to other types of flows, such as crowd flows, bike flows, and passenger flows of public transport.

Acknowledgments

The authors thank Prof. May Yuan, Prof. Grant McKenzie, and the anonymous reviewers for their insightful comments. The authors thank Dr. Junbo Zhang and M.S. Jie Li for providing the information of the experiment data.

Funding

This work is supported by the Science and Technology Project of Qingdao under Grant number 16-6-2-61-NSH and the CAS (Chinese Academy of Sciences) "100 Talent" Program (grant Y9KY04101L); The first author's joint Ph.D. research and the fifth author's Ph.D. research are funded by the China Scholarship Council (CSC). The CSC is a non-profit institution with legal person status affiliated with the Ministry of Education in China.

References

Abadi, M., *et al.* 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.

Bishop, C. M., 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc.

Caruana, R., Lawrence, S. and Giles, C. L., Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. ed. *Advances in neural information processing systems*, 2001, 402-408.

Chen, J., *et al.* 2018. Fine-grained prediction of urban population using mobile phone location data. *International Journal of Geographical Information Science*, 1-17.

Chen, P. T., Chen, F. and Qian, Z., Road Traffic Congestion Monitoring in Social Media with Hinge-Loss Markov Random Fields. ed. *IEEE International Conference on Data Mining*, 2015, 80-89.

Cheng, T. and Wang, J. 2008. Integrated Spatio-temporal Data Mining for Forest Fire Prediction. *Transactions in GIS,* 12(5), 591-611.

Cheng, T. and Wang, J. 2009. Accommodating spatial associations in DRNN for space–time analysis. *Computers, Environment and Urban Systems,* 33(6), 409-418.

Cheng, T., *et al.* 2014. A Dynamic Spatial Weight Matrix and Localized Space–Time Autoregressive Integrated Moving Average for Network Modeling. *Geographical Analysis,* 46(1), 75-97.

Cheng, T., Wang, J. and Li, X., Space-time series forecasting by artificial neural networks. ed. *International Conference on Earth Observation Data Processing and Analysis (ICEODPA)*, 2008, 72853I.

Chollet, F. 2015. Keras: Deep learning library for theano and tensorflow. *URL: https://keras. io/k,* 7, 8.

Gao, S., *et al.* 2013. Understanding Urban Traffic-Flow Characteristics: A Rethinking of Betweenness Centrality. *Environment and Planning B: Planning and Design,* 40(1), 135-153.

Gong, L., *et al.* 2016. Inferring trip purposes and uncovering travel patterns from taxi trajectory data. *Cartography and Geographic Information Science,* 43(2), 103-114.

Graves, A., Mohamed, A. R. and Hinton, G., Speech recognition with deep recurrent neural networks. ed. *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, 6645-6649.

Guo, D., *et al.* 2012. Discovering Spatial Patterns in Origin-Destination Mobility Data. *Transactions in GIS,* 16(3), 411-429.

He, K., *et al.*, Deep Residual Learning for Image Recognition. ed. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 27-30 June 2016 2016a, 770-778.

He, K., *et al.*, Identity mappings in deep residual networks. ed. *European Conference on Computer Vision*, 2016b, 630-645.

Heeren, T. and D'agostino, R. 1987. Robustness of the two independent samples t-test when applied to ordinal scaled data. *Statistics in medicine,* 6(1), 79-90.

Hinton, G. E. and Salakhutdinov, R. R. 2006. Reducing the dimensionality of data with neural networks. *Science,* 313(5786), 504-507.

Hoang, M. X., Zheng, Y. and Singh, A. K., FCCF: forecasting citywide crowd flows based on big data. ed. *The ACM Sigspatial International Conference*, 2016 Burlingame, California, 1-10.

Hochreiter, S. and Schmidhuber, J. 1997. Long short-term memory. *Neural Computation,* 9(8), 1735-1780.

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

Ke, J.*, et al.* 2017. Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach. *Transportation Research Part C: Emerging Technologies,* 85, 591-608.

Kingma, D. and Ba, J. 2014. Adam: A Method for Stochastic Optimization. *Computer Science*.

Lecun, Y., Bengio, Y. and Hinton, G. 2015. Deep learning. *Nature,* 521(7553), 436-444.

Lei, T., Zhang, Y. and Artzi, Y. 2017. Training rnns as fast as cnns. *arXiv preprint arXiv:1709.02755*.

Ma, X.*, et al.* 2017. Learning Traffic as Images: A Deep Convolutional Neural Network for Large-Scale Transportation Network Speed Prediction. *Sensors,* 17(4).

Ma, X.*, et al.* 2015. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transportation Research Part C: Emerging Technologies,* 54, 187-197.

Ma, Z.*, et al.* 2014. Predicting short-term bus passenger demand using a pattern hybrid approach. *Transportation Research Part C: Emerging Technologies,* 39, 148-163.

Nair, V. and Hinton, G. E., Rectified linear units improve restricted boltzmann machines. ed. *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, 807-814.

Shaw, S.-L., Tsou, M.-H. and Ye, X. 2016. Editorial: human dynamics in the mobile and big data era. *International Journal of Geographical Information Science,* 30(9), 1687-1693.

Shen, J. and Cheng, T. 2016. A framework for identifying activity groups from individual space-time profiles. *International Journal of Geographical Information Science,* 30(9), 1785-1805.

Shi, X.*, et al.* 2015. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. 802-810.

Silva, R., Kang, S. M. and Airoldi, E. M. 2015. Predicting traffic volumes and estimating the effects of shocks in massive transportation systems. *Proc Natl Acad Sci U S A,* 112(18), 5643-5648.

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

Wang, J.*, et al.*, STARIMA for journey time prediction in London. ed. *Proceedings of the 5th IMA conference on mathematics in transport*, 2010.

Wang, J., Cheng, T. and Li, X., Nonlinear Integration of Spatial and Temporal Forecasting by Support Vector Machines. ed. *International Conference on Fuzzy Systems and Knowledge Discovery*, 2007 Xi'an, China, 61-66.

Wang, J., Tsapakis, I. and Zhong, C. 2016. A space–time delay neural network model for travel time prediction. *Engineering Applications of Artificial Intelligence,* 52(C), 145-160.

Wang, Y., Currim, F. and Ram, S. 2017. Deep Learning for Bus Passenger Demand Prediction Using Big Data. *Social Science Electronic Publishing*.

Wu, Y. and Tan, H. 2016. Short-term traffic flow forecasting with spatial-temporal correlation in a hybrid deep learning framework. *arXiv preprint arXiv:1612.01022*.

Xingjian, S.*, et al.*, Convolutional LSTM network: A machine learning approach for precipitation nowcasting. ed. *Advances in neural information processing systems*, 2015, 802-810.

Zhang, J.*, et al.*, DNN-based prediction model for spatio-temporal data. ed. *ACM Sigspatial International Conference on Advances in Geographic Information Systems*, 2016, 92.

Zhang, J.*, et al.* 2018. Predicting Citywide Crowd Flows Using Deep Spatio-Temporal Residual Networks ☆. *Artificial Intelligence*.

Zhang, L. and Weghe, N. V. D. 2018. Attribute trajectory analysis: a framework to analyse attribute changes using trajectory analysis techniques. *International Journal of Geographical Information Science,* (2), 1-17.

Zheng, Y.*, et al.* 2014. Urban Computing:Concepts, Methodologies, and Applications. *Acm Transactions on Intelligent Systems & Technology,* 5(3), 1-55.

Zhu, D.*, et al.* 2018. Inferring spatial interaction patterns from sequential snapshots of spatial distributions. *International Journal of Geographical Information Science,* 32(4), 783-805.

Zhu, X. and Guo, D. 2014. Mapping Large Spatial Flow Data with Hierarchical Clustering. *Transactions in GIS,* 18(3), 421-435.

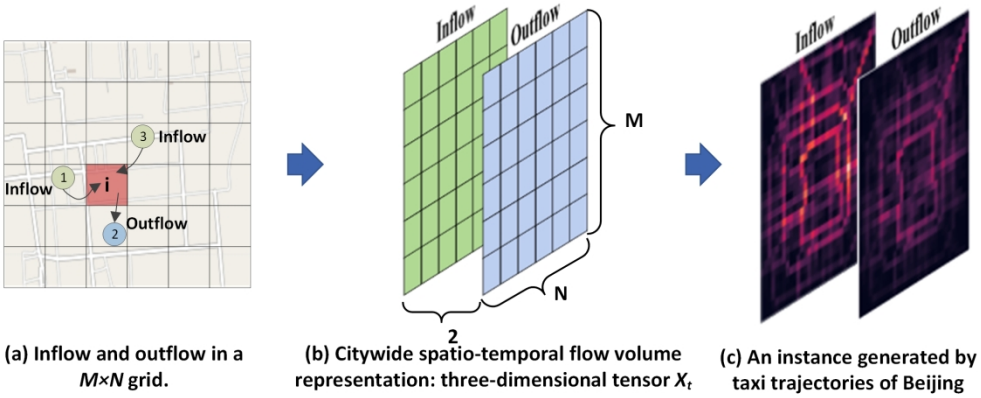

Figure 1. Citywide spatio-temporal flow volume
The citywide spatio-temporal flow volume quantitatively reflects the

194x80mm (300 x 300 DPI)

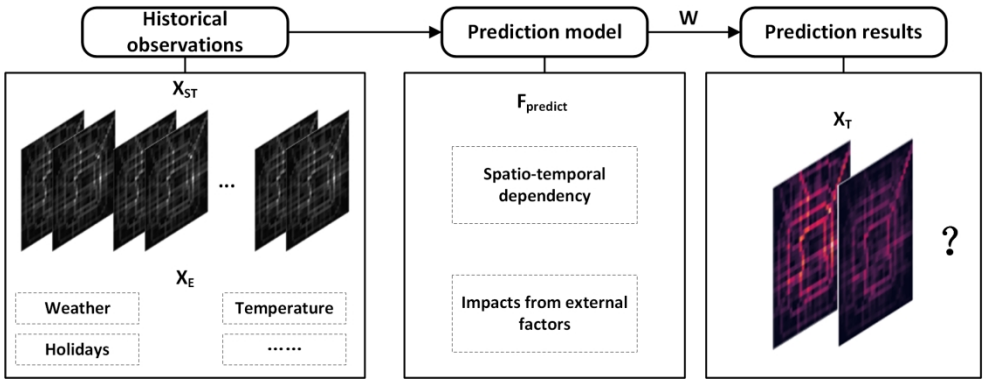

Figure 2. Problem definition of citywide spatio-temporal flow volume prediction

194x74mm (300 x 300 DPI)

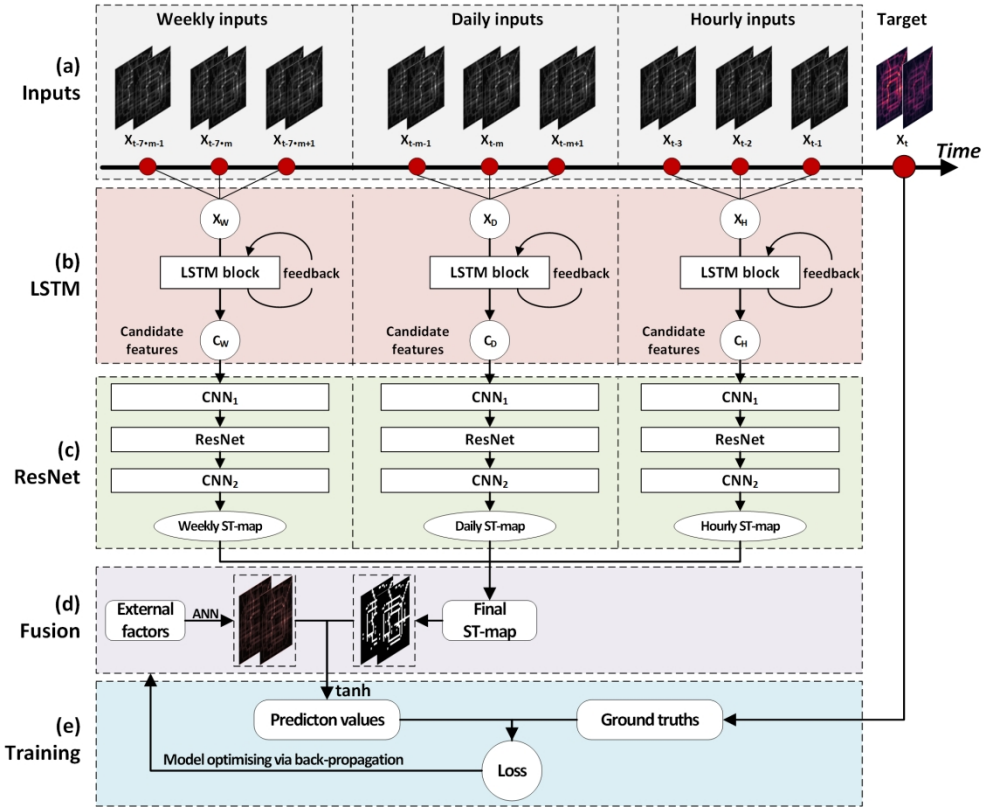

Figure 3. Framework of HIDLST

188x151mm (300 x 300 DPI)

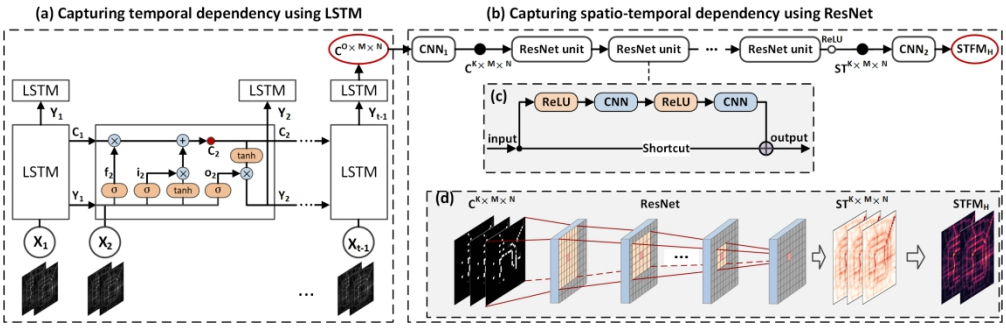

Figure 4. Integrally capturing the spatio-temporal dependency

212x68mm (300 x 300 DPI)

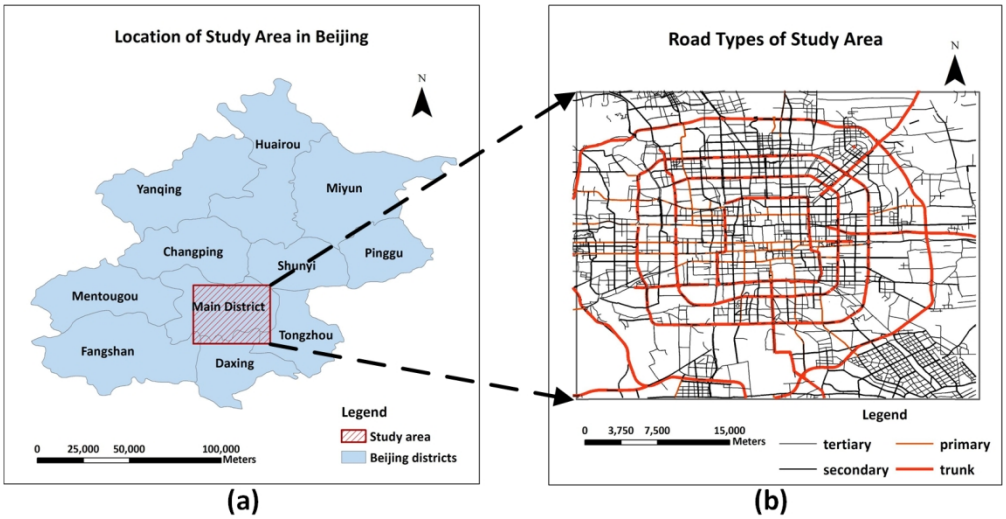

Figure 5. Study area

141x74mm (300 x 300 DPI)

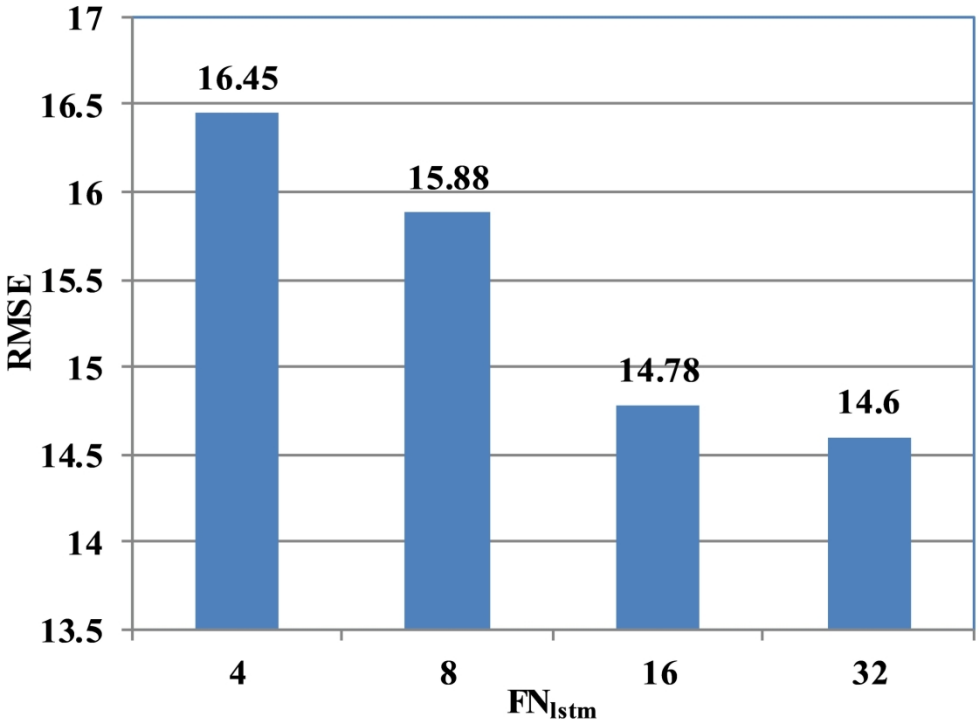

Figure 6. Experimental results for different hidden neuron units

98x73mm (600 x 600 DPI)

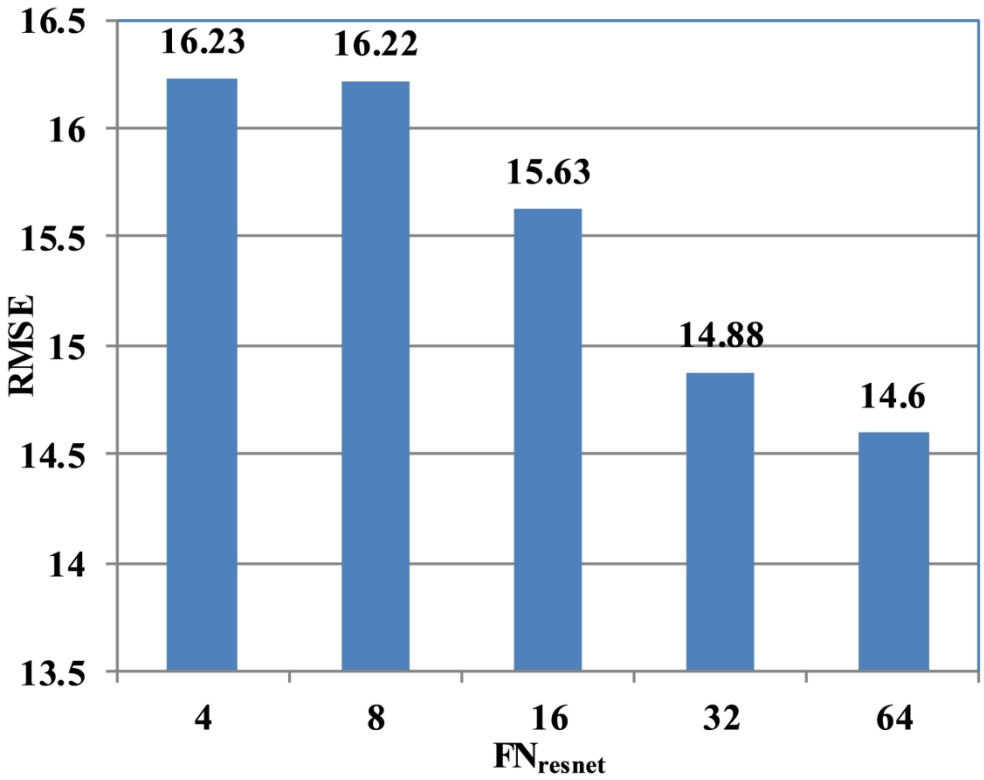

Figure 6. Experimental results for different hidden neuron units

94x73mm (600 x 600 DPI)

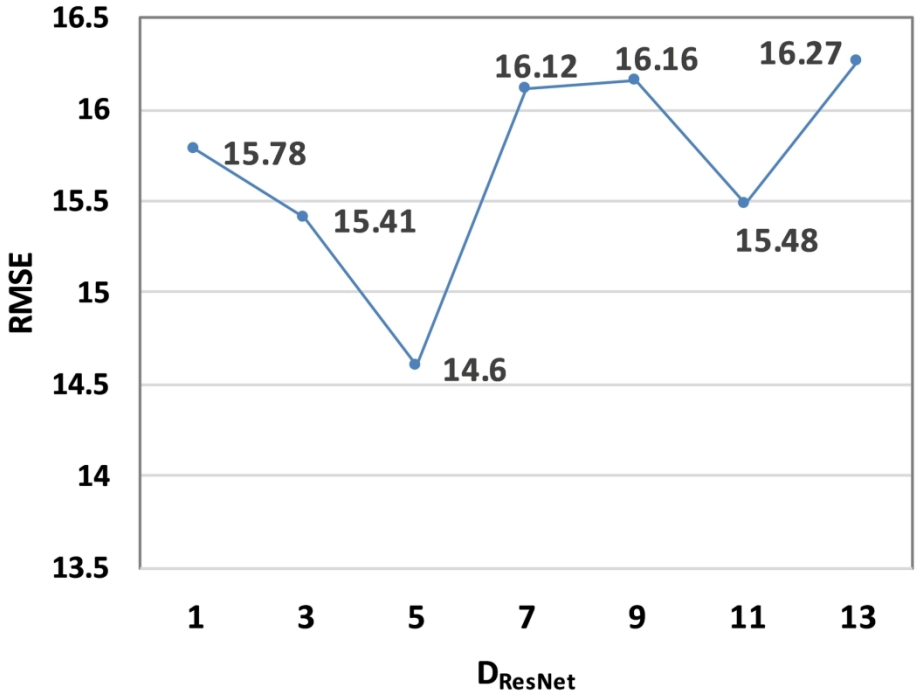

Figure 7. Experimental results for different model depths

97x73mm (600 x 600 DPI)

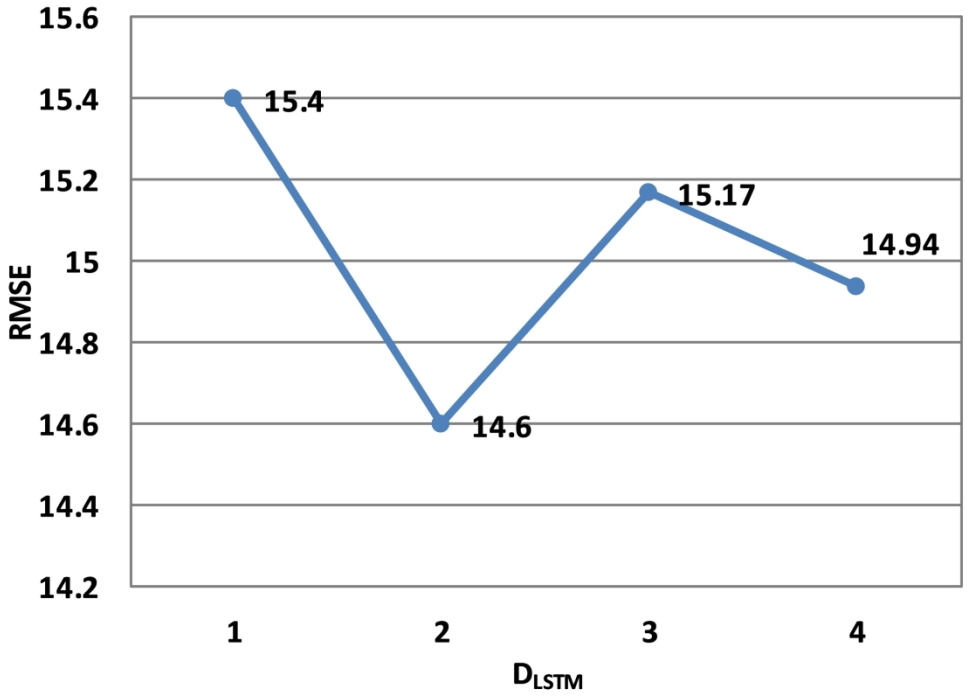

Figure 7. Experimental results for different model depths

102x73mm (600 x 600 DPI)

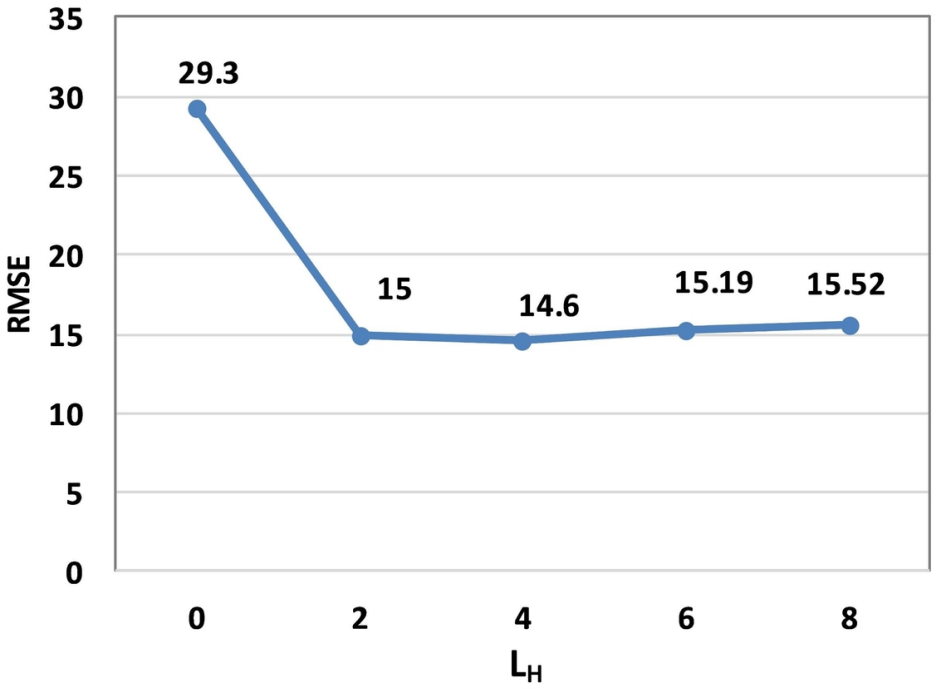

Figure 8. Experimental results for different input lengths

101x76mm (300 x 300 DPI)

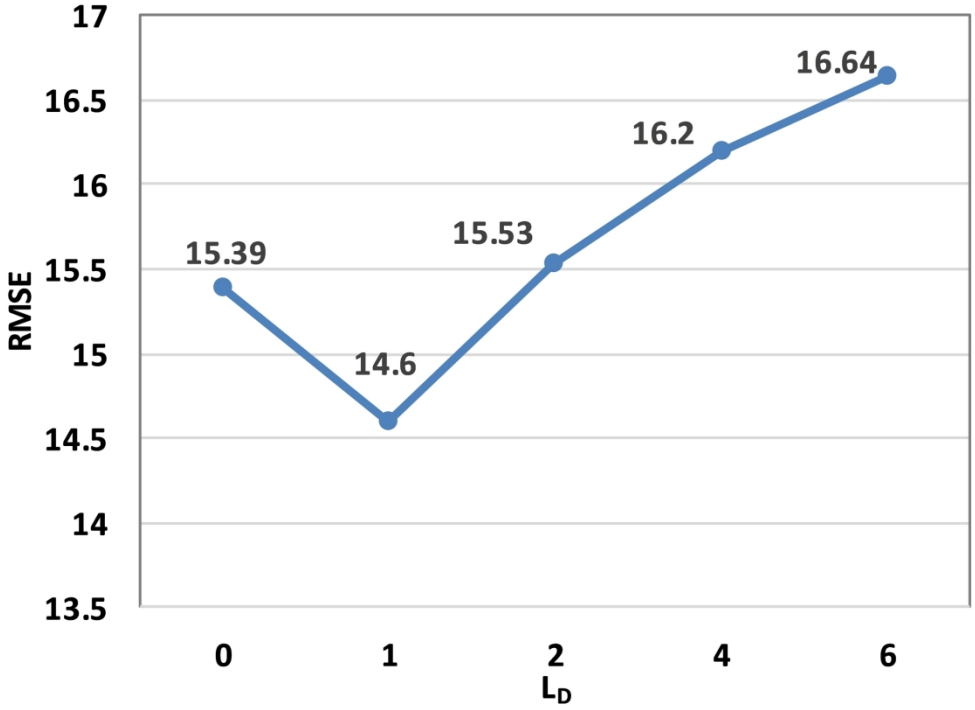

Figure 8. Experimental results for different input lengths

99x73mm (600 x 600 DPI)

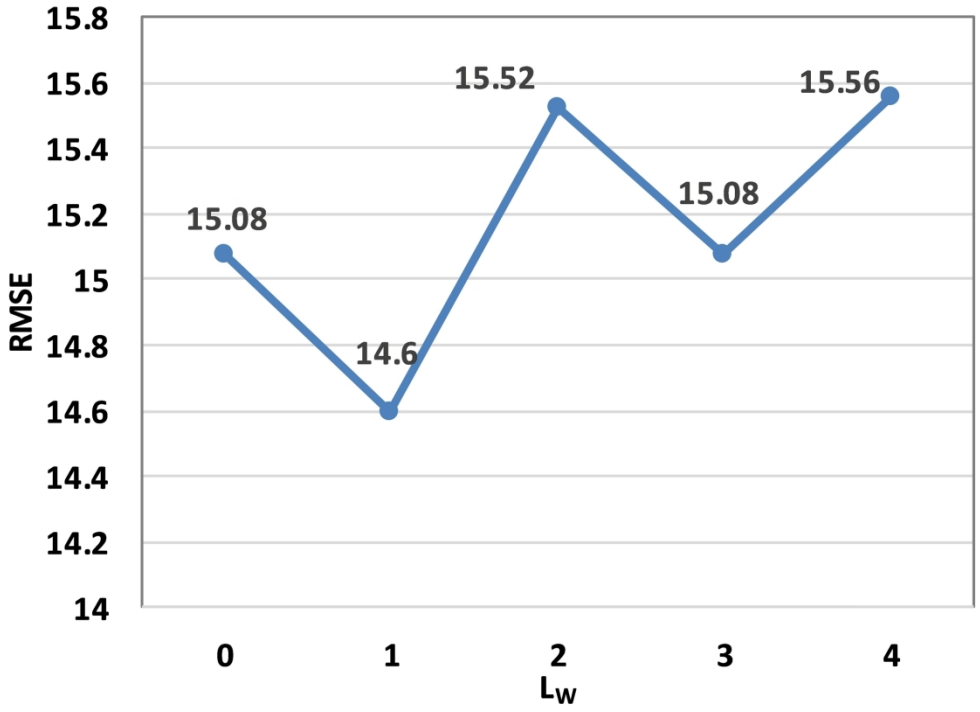

Figure 8. Experimental results for different input lengths

99x73mm (600 x 600 DPI)

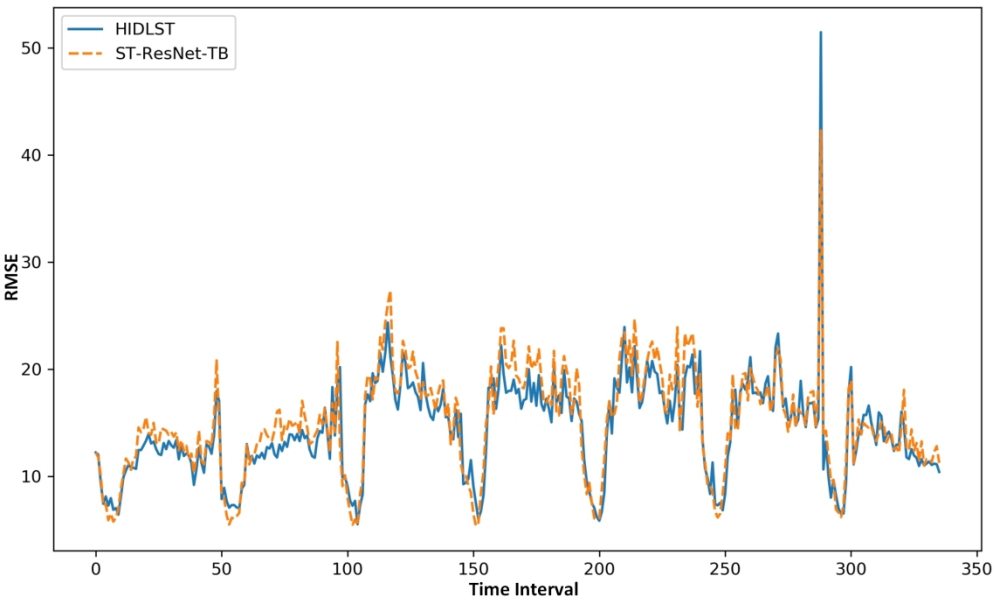

Figure 9. RMSEs of HIDLST and ST-ResNet-TB during of all predicted time intervals

211x128mm (300 x 300 DPI)

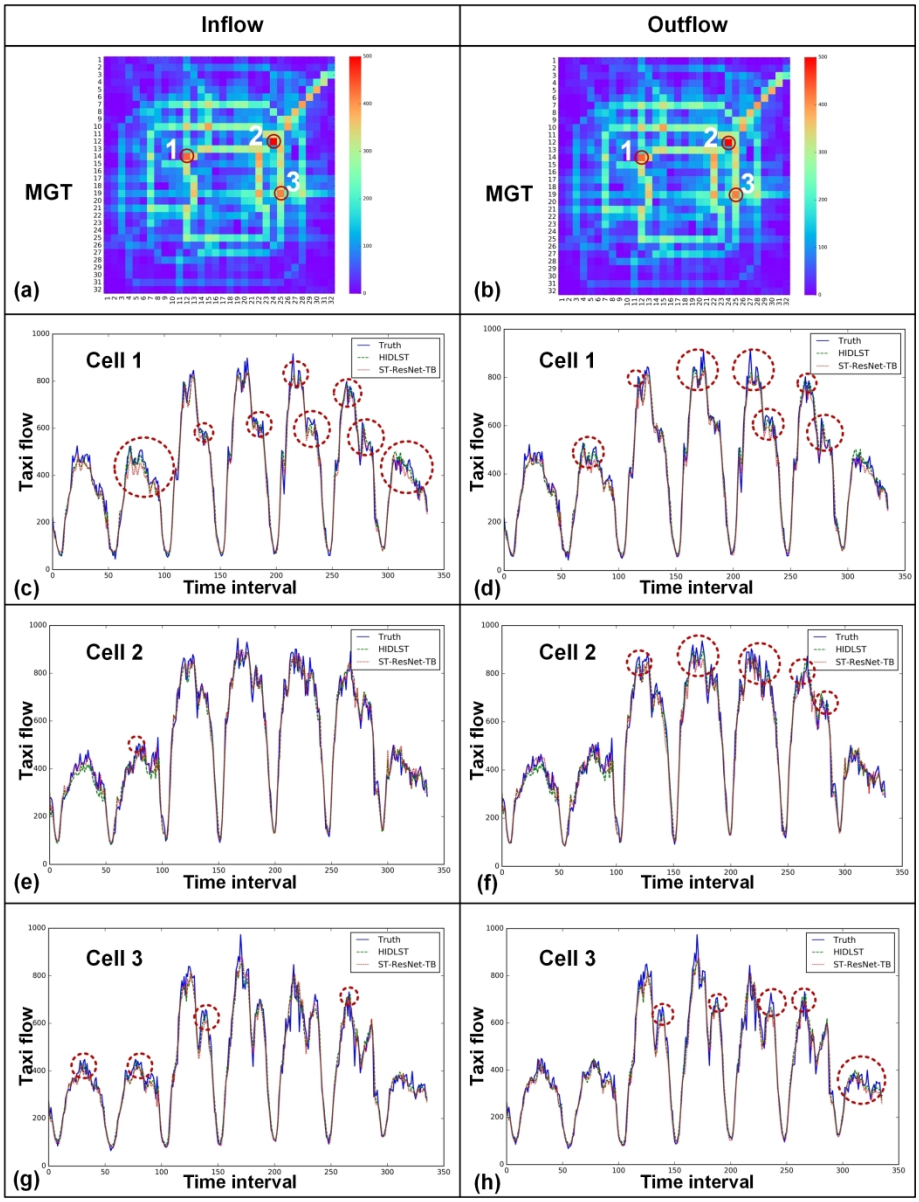

Figure 10. Comparisons about ground truths, predictions of HIDLST and predictions of ST-ResNet-TB

208x272mm (300 x 300 DPI)

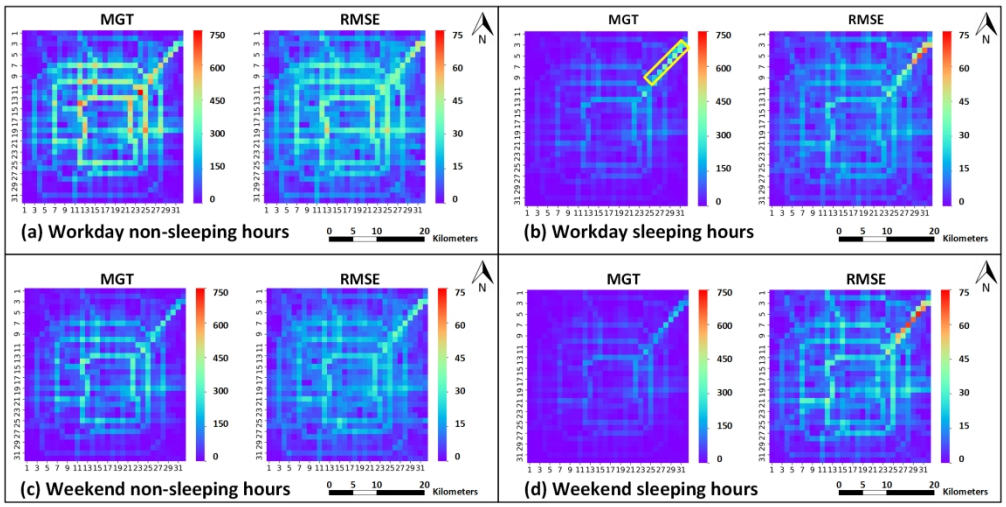

Figure 11. Distribution of MGT and RMSE

209x106mm (300 x 300 DPI)

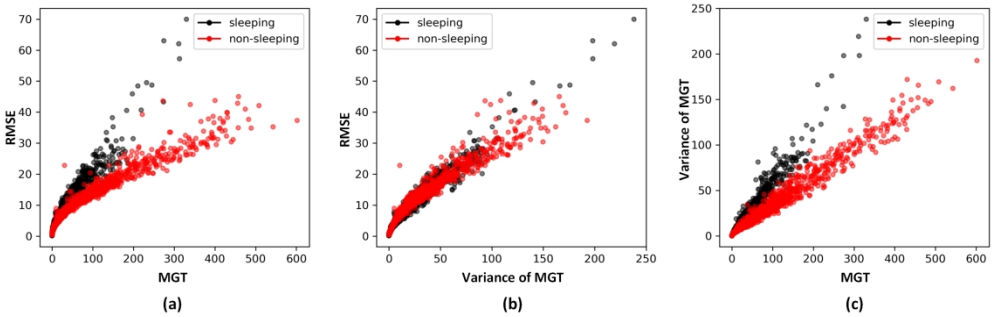

Figure 12. Relationships of MGT and RMSE, RMSE and variance of MGT, and variance of MGT and MGT

239x76mm (300 x 300 DPI)

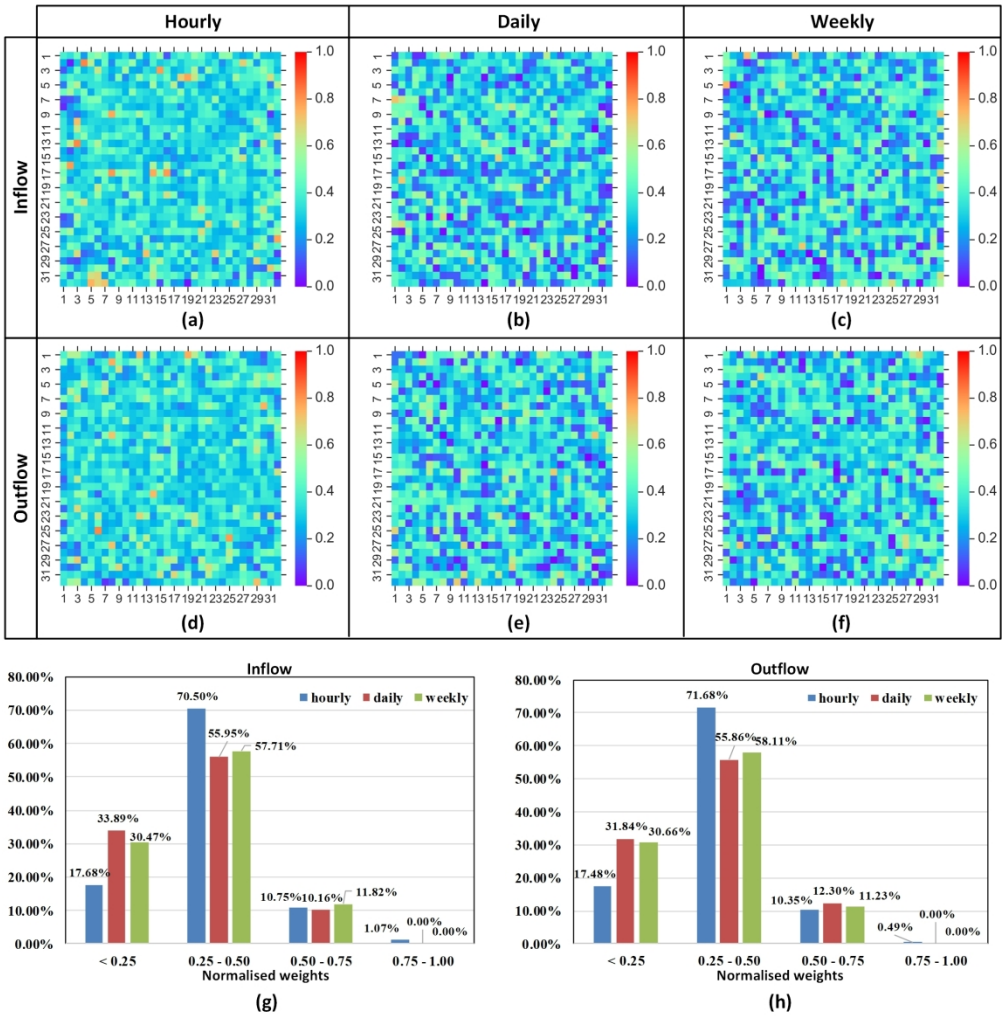

Figure 13. Weight matrices' distribution of hourly, daily and weekly patterns

205x210mm (300 x 300 DPI)