Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 4173094, 10 pages
https://doi.org/10.1155/2020/4173094

*Research Article*

# Multitime Resolution Hierarchical Attention-Based Recurrent Highway Networks for Taxi Demand Prediction

**Baiping Chen and Wei Li**

*Institute of Intelligent and Software Technology, Hangzhou Dianzi University, Hangzhou 310018, China*

Correspondence should be addressed to Wei Li; lw@hdu.edu.cn

Received 15 June 2020; Revised 1 August 2020; Accepted 3 August 2020; Published 20 August 2020

Academic Editor: Ricardo Aguilar-Lopez



Taxi demand forecasting is an important consideration in building up smart cities. However, complex nonlinear spatiotemporal relationships in demand data make it difficult to construct an accurate prediction model. Considering that a single time resolution may not enable accurate learning of the time pattern of taxi demand, we expand the time series prediction model in our proposed multitime resolution hierarchical attention-based recurrent highway network (MTR-HRHN) model, using three time resolutions to model temporal closeness, period, and trend properties of demand data to capture a more comprehensive time pattern. We evaluate the MTR-HRHN on a taxi trip record dataset and the results show that the forecasting performance of the MTR-HRHN exceeds that of eight well-known methods in the short-term demand prediction in some high-demand regions.

## 1. Introduction

With the increasing travel demand of urban dwellers, taxis have become much more popular in urban areas, especially through the use of ride hailing services such as Didi Chuxing and Uber. However, the business still faces many inefficiencies, including long waits and numerous empty taxis [1–3]. The use of data technology and artificial intelligence to process massive taxi data can enable the construction of an accurate prediction model that can be used to estimate taxi demand and improve the efficiencies of taxi services. For example, the number of passengers from different regions was predicted [4–6] through a linear time series model. The impact of the road network and meteorological conditions on the demands of taxis was researched [7, 8] using a method of machine learning. For the demand forecasting problem, the common method for taxi demand prediction is to consider the impact of historical demand data on future demand; that is, predict demand $\hat{y}_T$ at time $T$, given a series of historical demands $(y_1, y_2, \ldots, y_{T-1})$. The time interval $T$ is a short-term time, which is often a few hours or even shorter. However, for data such as taxi demand with nonlinear, unstable, and spatiotemporal related properties, linear or nonlinear methods considering only historical demand are insufficient. The following points should be considered when constructing the prediction model:

(1) Besides historical demand data, relevant exogenous data are necessary and should be applied to train the model. In this regional forecasting problem, exogenous data are often selected from other regions.

(2) The model should be nonlinear and should consider not only the temporal dependence of target data and exogenous data but the relationship between target data and other exogenous data.

Figure 1 is a spatiotemporal dynamic structure that models both the historical target data and historical exogenous data. As pointed out in [9], $\hat{y}_T$ is related to the historical observations $(y_1, y_2, \ldots, y_{T-1})$, the exogenous data $(x_1, x_2, \ldots, x_{T-1})$, and their spatiotemporal dynamics. For their excellent performance in learning the dynamic dependence in sequences, deep learning models, such as the recurrent neural network (RNN) and its extended variants, have been used to capture the nonlinear temporal

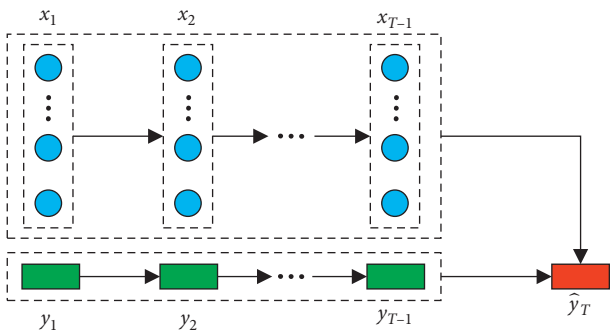


Figure 1: Spatiotemporal dynamic model of historical observations and exogenous data.

relationships of time series data. In addition, the convolutional neural network (CNN) can be added to capture the spatial correlation [10]. The encoder-decoder architecture was recently used to model sequence data [11, 12], and some attention-based models [13] have been proposed to exploit the temporal dynamics of exogenous data when predicting future targets. However, these models do not consider the correlation of exogenous data between different components and the time factor in series data, and this will affect the prediction results. Overcoming these issues is the motivation of our research.

In this paper, we extend a hierarchical attention-based recurrent highway network (HRHN) [9] and propose a multitime resolution model, MTR-HRHN. We select different lengths of sequence data from historical time series data (including target data and exogenous data) with three different time resolutions and input the sampled data to three HRHN networks to train the model to capture the spatiotemporal characteristics. We merge the output of each HRHN network to predict taxi travel demand at a certain time in the region. Compared with other spatiotemporal deep learning network models, our network has the ability to learn from three time resolutions. It can not only extract the spatiotemporal characteristics of time series data and their relationship with exogenous data, but can capture the influence of recent, periodic, and trend factors on taxi demand.

The organization of this paper is as follows. A brief overview of traditional prediction methods and deep learning models in traffic data prediction is given, followed by some definitions of demand prediction. The structure of MTR-HRHN is then described. We test the MTR-HRHN model on the New York City taxi dataset and compare it to other models. In the conclusion, we summarize the paper and provide some inspiration for improving the model.

## 2. Related Work

Statistics-based algorithms (such as ARIMA and its variants) [4, 5, 7] and machine learning regression models (such as linear regression and support vector machine) [6–8] are widely used in the research of traffic prediction. However, in the real world, the demand data of a certain region are often affected by other nonnumeric data (such as changes in weather), which prevents the linear model from completely digging out relevant information.

Recent superior performance of deep learning in computer vision and natural language processing has encouraged its application to traffic data prediction. Among them, the CNN can strongly extract the features of the input data, so it is naturally used for traffic prediction [14–16]. The RNN and some of its extended variants, such as LSTM [17] and the gated recurrent unit (GRU) [18], are outstanding at capturing dynamic time dependence and are widely used to predict time series data [19–23]. For example, Xu et al. encoded past taxi demand into week-long sequences, fed the sequential data to an LSTM network, and made the network learn the taxi demand patterns in each area. Rather than forecasting a deterministic taxi demand, it predicted the entire probability distribution of taxi demand in different areas through mixture density networks [22]. However, when dealing with regional demand prediction, different regions relate to each other, and the demand change of a certain region often has a certain correlation with the demand data of other regions. The inability to simultaneously capture spatial and temporal relations made these deep learning models inapplicable to our problem.

Therefore, some researchers have chosen to build spatiotemporal deep learning models for traffic data prediction [10, 24, 25]. Among them, the combined deep network of CNN and LSTM is a classic spatiotemporal deep learning model. For example, Yao et al. proposed a novel local CNN method to consider spatial near regions and extract the sequential relations in a demand time series, and some LSTM networks were used to model sequential dependencies [10]. The encoder-decoder framework was also used by some researchers to deal with the spatiotemporal relationships of traffic data [24, 25]. For example, Zhou et al. proposed an encoder-decoder framework with attention mechanism to deal with the multistep citywide passenger demand prediction problem. They employed convolutional and ConvLSTM units in both the encoder and decoder and learned attention to emphasize the effects of representative citywide demand patterns on each step prediction during the decoding phase [24]. Some studies have expanded the spatiotemporal models to solve some traffic prediction problems that require more precision. For example, Rodrigues et al. proposed a deep learning architecture combining text information with time-series data and applied the approach to the problem of taxi demand forecasting in event areas [26]. Liu et al. proposed a contextualized spatial-temporal network to deal with the taxi origin-destination problem, integrating the local spatial context, temporal evolution context, and global correlation context in a united framework [27]. Although these spatiotemporal deep networks showed outstanding performance in the transportation field, they have some shortcomings, as they only sample historical traffic data from a single time resolution (such as a half hour or hour), which may lead to the inability to fully mine the possible multitime patterns of traffic data.

Apart from the above spatiotemporal models, HRHN, as an end-to-end deep learning model, has the ability to predict

future target data by mining the spatial and temporal interaction information of historical exogenous and target data. It has been tested in several domains and proved able to not only achieve accurate prediction of time series but to better capture their sudden changes and oscillations [9]. Inspired by the capabilities of HRHN in the prediction of time series data, we chose it to learn the spatial and temporal correlation information between the demand data of the target region and the demand data of other regions. Moreover, to adapt to possible multitime patterns in demand data, unlike the original HRHN model, our model uses three time resolutions to sample the past target demand data and demand data of related regions and feeds them to three HRHN models to extract the corresponding spatiotemporal correlation information.

## 3. Definitions

*3.1. Trip.* A trip is defined as a tuple, trip = $(\mathrm{id}, t_{\mathrm{start}}, l_{\mathrm{start}}, t_{\mathrm{end}}, l_{\mathrm{end}})$, where $i \quad d$ is the trip identification number, $t_{\mathrm{start}}$ and $l_{\mathrm{start}}$ are, respectively, the time and place a passenger gets on a taxi, and $t_{\mathrm{end}}$ and $l_{\mathrm{end}}$ are, respectively, the time and place the passenger gets off the taxi.

*3.2. Taxi Demand.* For a region $i$, the taxi pick-up demands $D_{i,j}^{p}$ generated in the time interval $[T_j, T_{j+1})$ are defined as

$$D_{i,j}^{p} = \left| \left\{ \mathrm{trip} \,\middle|\, t_{\mathrm{start}} \in \left[T_j, T_{j+1}\right) \wedge l_{\mathrm{start}} = i \right\} \right|. \tag{1}$$

*3.3. Short-Term Demand Prediction Problem.* In this study, we set the length of each time interval to one hour and only predict the demand data of the selected region in a specific future time. For a fixed region $i$ and time interval $T$, the one-step demand prediction problem can be defined as follows: given a series of historical demand data $(D_{i,T-h}^{p}, D_{i,T-h+1}^{p}, \ldots, D_{i,T-1}^{p})$ and related historical exogenous data $(x_{T-h}, x_{T-h+1}, \ldots, x_{T-1})$, the task is to predict the demand value of this region at future time interval $T$:

$$D_{i,T}^{p} = F\left(D_{i,T-h}^{p}, D_{i,T-h+1}^{p}, \ldots, D_{i,T-1}^{p}, x_{T-h}, x_{T-h+1}, \ldots, x_{T-1}\right), \tag{2}$$

where $D_{i,t}^{p} \in R^{1}\,(T-h \le t \le T-1)$ represents the pick-up demand at a given region $i$ at time $t$, $h$ is the length of the input sequence data, $x_i \in R^{e}\,(T-h \le i \le T-1)$ is the exogenous data and $e$ is its dimension, and $F(\cdot)$ is a function to be learned that captures the complex spatiotemporal interaction between historical target and exogenous data.

## 4. Methods

As shown in Figure 2, MTR-HRHN has three layers: input, HRHN, and merge. In the input layer, we divide the historical target and exogenous data according to three time resolutions and select different lengths of sequence data to form the recent-, near- and distant-time training samples. To match the time characteristics of the three HRHN networks, the time resolution of the recent-time samples is the smallest, followed by near-time samples and then distant-time samples.

In the HRHN layer, three HRHN networks train the model from three time-related perspectives: recent, period, and trend. Each HRHN network has an exogenous data capture part ($X$) and a demand forecast part (DP). Each $X$ is linked to a sequence of historical exogenous data, and each DP is linked to a sequence of historical target data. The attention mechanism of the HRHN further learns the association between the target and exogenous data.

In the merge layer, the output of each HRHN undergoes the transformation of the fully connected layer. The transformed data are summed to obtain the final demand prediction data. The prediction data are used to construct a loss function together with the real data, and the model parameter training is completed through an optimization algorithm.

MTR-HRHN has an encoder-decoder structure and the ability to process sequence learning. Unlike most spatiotemporal deep network learning models that use LSTM, our model uses RHN to capture the temporal feature and embeds RHN in both the encoder and decoder. Compared to LSTM, RHN can offer a deeper understanding of the strengths of the LSTM cell and incorporate highway layers inside the recurrent transition, enabling the efficient use of substantially more powerful and trainable sequential models [28]. To our knowledge, HRHN has not been used in the field of taxi demand forecasting. For this new application, we employed a new model with multiple HRHNs, and the input layer, merge layer, and training algorithm are designed accordingly, so that the expanded new model has a better ability to learn spatiotemporal correlation sequences.

*4.1. Input Layer.* We use three time resolutions to divide the historical demand data and historical exogenous data into three parts: closeness, period, and trend. The recent historical exogenous data $X_{i,r} = (x_{T-L_c}, \ldots, x_{T-2}, x_{T-1})$ and recent historical target demand data $\mathrm{DP}_{i,r} = (D_{i,T-L_c}^{p}, \ldots, D_{i,T-2}^{p}, D_{i,T-1}^{p})$ are selected for the closeness part, where $L_c$ is the number of time intervals of the closeness fragment. The near historical exogenous data $X_{i,n} = (x_{T-L_p\times P}, \ldots, x_{T-2\times P}, x_{T-P})$ and near historical target demand data $\mathrm{DP}_{i,n} = (D_{i,T-L_p\times P}^{p}, \ldots, D_{i,T-2\times P}^{p}, D_{i,T-P}^{p})$ are selected for the period part, where $L_p$ is the number of time intervals of the period fragment. The distant historical exogenous data $X_{i,d} = (x_{T-L_t\times Q}, \ldots, x_{T-2\times Q}, x_{T-Q})$ and distant historical target demand data $\mathrm{DP}_{i,d} = (D_{i,T-L_t\times Q}^{p}, \ldots, D_{i,T-2\times Q}^{p}, D_{i,T-Q}^{p})$ are selected for the trend part, where $L_t$ is the number of time intervals of the trend fragment. It is noted that $P$ and $Q$ are different types of periods, where $P$ is equal to 12 and reveals the half-daily periodicity, and $Q$ is equal to 24 and reveals the daily trend.

*4.2. HRHN Layer.* We applied the HRHN [9] to the regional demand prediction problem. The CNNs in the encoder

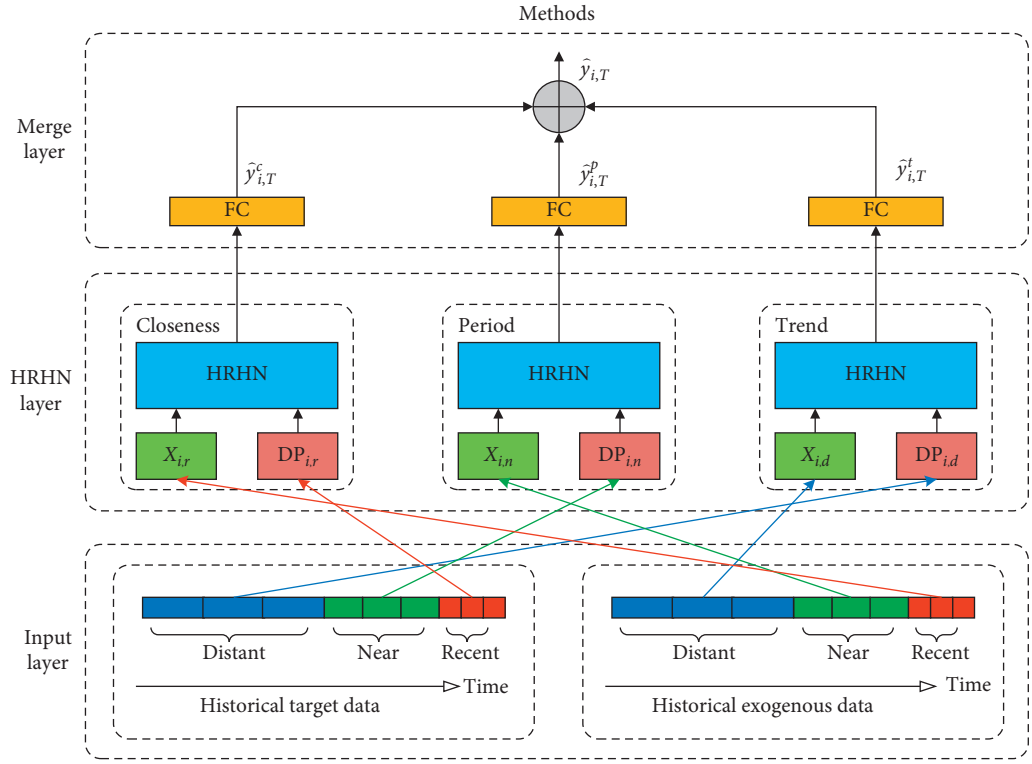


Figure 2: The structure of our multitime resolution hierarchical attention-based recurrent highway network (MTR-HRHN).

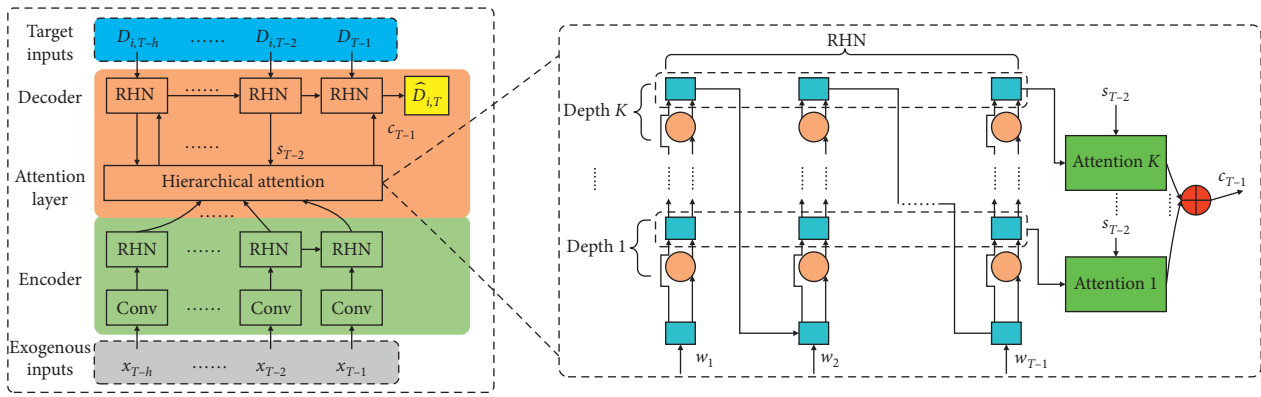


Figure 3: The structure of the hierarchical attention-based recurrent highway network (HRHN).

learn spatial-related information from different components of demand data of other related regions, and RHNs in the encoder model and analyze the temporal dependence of demand data of related regions from the CNN at different semantic levels. RHNs in the decoder capture the time-dependent information of the historical demand of the region to be predicted. The decoder also includes a hierarchical attention mechanism, so that it can select the relevant multilevel semantic encoded information.

*4.2.1. Encoder.* Convolutional neural networks and pooling layers are used in the encoder to learn spatial information from components of exogenous data. Suppose the number of convolutional network layers corresponding to each moment is $K_c$, and the number of feature maps of the $u$-th layer is $F_u$. Assuming that the kernel size of each convolutional layer is set as $1 \times q$, then the $i$-th convolution unit of the $f$-th feature map of the $u$-th layer can be calculated from the data of the $u-1$ layer as

$$x_{(u,f)}^{i} = \mathrm{ReLU}\left(\sum_{p=1}^{F_u}\sum_{j=0}^{q-1} k_{(u,f,j)} x_{(u-1,p)}^{i+j} + b_{(u,f)}\right), \quad 1 \le u \le K_c - 1, \tag{3}$$

where $k_{(u,f,i)}$ is the $j$-th unit of the convolution kernel of the $f$-th channel graph of the $u$-th layer and $b_{(u,f)}$ is the bias term. In addition, for layer 1 (in this case, $u = 1$), the input data are exogenous; that is, when $u = 1$, $x_{(u-1,p)}^{i+j} = x_{(0,1)}^{i+j} = x_{i+j}$. The maximum pooling immediately following the convolution operation is

$$x_{(u+1,f)}^{k} = \max\left(x_{(u,f)}^{sk}, x_{(u,f)}^{sk}, \ldots, x_{(u,f)}^{sk+s-1}\right), \tag{4}$$

where $s$ is the size of the maximum pooling layer.

After processing by the $K_c$ layers of the convolutional and pooling layers, the local feature vector $(w_1, w_2, \ldots, w_{T-1})$ can be obtained.

The RHN in the encoder analyzes the temporal dependence of the input data from the CNN. The relevant equations are as follows:

$$\begin{aligned} h_t^{[k]} &= g_t^{[k]} \cdot r_t^{[k]} + h_t^{[k-1]} \cdot c_t^{[k]}, \\ g_t^{[k]} &= \tanh\left(W_G w_t I\{k=1\} + V_{G_k} h_t^{[k-1]} + b_{G_k}\right), \\ r_t^{[k]} &= \sigma\left(W_R w_t I\{k=1\} + V_{R_k} h_t^{[k-1]} + b_{R_k}\right), \\ c_t^{[k]} &= \sigma\left(W_C w_t I\{k=1\} + V_{C_k} h_t^{[k-1]} + b_{C_k}\right), \end{aligned} \tag{5}$$

where $I$ is an indicator function, $h_t^{[k]}$ is the intermediate output at time $t$ and depth $k$ in RHN, and $I\{k=1\}$ means that $w_t$ only participates in the transformation at the first layer. In addition, the first layer network $h_t^{[k-1]} \in R^l$ corresponds to the output data of the last layer at time $t-1$.

*4.2.2. Decoder.* The decoder contains another RHN used to capture the time-dependent information of the historical demand sequence data of the region to be predicted. An attention mechanism is introduced to solve the problem of encoding longer input sequences.

An attention model was originally used for machine translation [29] and has been widely used in natural language processing, statistical learning, speech, and the computer fields. A hierarchical attention mechanism, which performs better than the traditional attention mechanism, was developed based on the original attention model. For example, when processing document classification, the hierarchical attention mechanism can simultaneously build sentence- and word-level attention models, while the traditional attention mechanism can only construct a single level of attention model.

The decoder of HRHN introduces a hierarchical attention mechanism, which can mine the information stored in different layers to capture temporal dynamics at different levels, which will have a better impact on predicting future target series compared to the traditional attention mechanism [9]. The alignment model $e_{t,i}^{[k]}$ is calculated as follows:

$$\alpha_{t,i}^{[k]} = \frac{e_{t,i}^{[k]}}{\sum_{j=1}^{T-1} e_{t,j}^{[k]}}, \quad 1 \le i \le T-1, \tag{6}$$

where

$$e_{i,j}^{[k]} = v_k^T \tanh\left(T_k s_{t-1} + U_k h_i^{[k]}\right). \tag{7}$$

$s_{t-1} = s_{t-1}^{[K_r]} \in R^p$ represents the output of the last layer of RHN in the decoder at time *t-1*, and $v_k \in R^l$, $T_k \in R^{e \times p}$, and $v_k \in R^{l \times l}$ are all trainable parameters.

By computing the subcontext vector $d_t^{[k]}$ as a weighted sum of all the encoder's hidden states in the $k$-th layer, the soft alignment for layer $k$ is obtained as

$$d_t^{[k]} = \sum_{i=1}^{T-1} \alpha_{t,i}^{[k]} h_i^{[k]}. \tag{8}$$

Then, the context vector that we feed to the decoder is calculated as

$$d_t = \left[d_t^{[1]}, d_t^{[2]}, \ldots, d_t^{\left[K_r\right]}\right], \tag{9}$$

where $K_r$ is the number of RHN layers.

From the output of the encoder to the input of the decoder, $\widetilde{D}_t^p$ is a time-dependent variable representing the interaction between $D_{i,t}^p$ and $d_t$:

$$\widetilde{D}_t^p = \widetilde{W} D_{i,t}^p + \widetilde{V} d_t + \widetilde{b}, \tag{10}$$

where $\widetilde{W} \in R^{1\times 1}$ and $\widetilde{V} \in R^{1 \times K_r e}$ are the weight matrices and $\widetilde{b} \in R^1$ is the bias term.

RHN in the decoder is similar to that in the encoder, with the following related equations:

$$\begin{aligned} s_t^{[k]} &= \widetilde{g}_t^{[k]} \cdot \widetilde{r}_t^{[k]} + \widetilde{s}_t^{[k-1]} \cdot \widetilde{c}_t^{[k]}, \\ \widetilde{g}_t^{[k]} &= \tanh\left(\widetilde{W}_G \widetilde{D}_t^p I\{k=1\} + \widetilde{V}_{G_k} h_t^{[k-1]} + \widetilde{b}_{G_k}\right), \\ \widetilde{r}_t^{[k]} &= \sigma\left(\widetilde{W}_R \widetilde{D}_t^p I\{k=1\} + \widetilde{V}_{R_k} h_t^{[k-1]} + \widetilde{b}_{G_R}\right), \\ \widetilde{c}_t^{[k]} &= \sigma\left(\widetilde{W}_C \widetilde{D}_t^p I\{k=1\} + \widetilde{V}_{C_k} h_t^{[k-1]} + \widetilde{b}_{G_C}\right), \end{aligned} \tag{11}$$

where $\widetilde{W}_{G,R,C} \in R^{p\times 1}$ and $\widetilde{V}_{G,R,C} \in R^{p\times p}$ represent the transformation functions of the nonlinear transformation $G$, transformation gate $R$, and carry gate C and $\widetilde{b}_{G,R,C} \in R^p$ are bias terms.

The estimated value $\widehat{D}_{i,t}^p$ of the pick-up demand in time interval $T$ of the region $i$ to be predicted under this time mode can be obtained as

$$\widehat{D}_{i,T}^p = W s_{T-1}^{\left[K_r\right]} + V d_{T-1} + b, \tag{12}$$

where $s_{T-1}^{[K_r]}$ is the output data of the last layer of RHN in the decoder and $d_{T-1}$ is the associated context vector. The parameters $W \in R^{1\times p}$, $V \in R^{1\times K_r l}$, and $b \in R^1$ are trainable parameters that characterize the linear dependence and produce the final prediction.

*4.3. Merge Layer.* The historical demand data and the historical exogenous data of the closeness, period, and trend parts are fed to the HRHNs. Then we multiply each output of HRHN with the corresponding weight matrix and add the results together to get the final prediction data:

$$\widehat{D}_{i,T}^p = W_f^c \widehat{D}_{i,T}^{p,c} + W_f^p \widehat{D}_{i,T}^{p,p} + W_f^t \widehat{D}_{i,T}^{p,t}, \tag{13}$$

where $W_f^{c,p,t} \in R^1$ are trainable weight matrices.

*4.4. Loss Function and Optimizer.* After obtaining the predicted data $\widehat{D}_{i,T}^p$, the mean square error is used as the loss function of the model:

$$\text{Loss}_i = \sum_{n=1}^{N} \left(\widehat{D}_{i,n}^p - D_{i,n}^p\right)^2, \tag{14}$$

where $N$ is the number of training data points and $\widehat{D}_{i,n}^{p}$ and $D_{i,n}^{p}$ represent the predicted demand and real demand data, respectively, of region $i$ at time interval $n$. $\text{Loss}_i$ is the loss function of the pick-up demand forecast for the region $i$.

In addition, each region has an independent loss function. The model uses the Adam optimizer to complete the training [30]. During the training process, the output of the loss function of the validation set is calculated in each iteration. If the value is less than the minimum value of the previous iterations, then the parameters of MTR-HRHN at this iteration are saved and the value is updated as the new minimum value. The termination condition of training is when the value of the loss function of the validation set corresponding to several consecutive iterations is greater than or equal to the minimum value.

## 5. Results and Discussion

The dataset selected for the experiment was the New York City Yellow Taxi Trip Records (https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page) from January 1 to March 31, 2019.

Regarding the region division, there are many methods to divide cities into regions with different granularities and semantic meanings, such as road networks and ZIP code tabulation areas [31]. We used the New York City regional division scheme attached to the dataset to divide the city into six regions: The Bronx, Brooklyn, EWR, Manhattan, Queens, and Staten Island. We selected 12 high-demand subregions from Manhattan as the experimental objects shown in Table 1. We selected data from the last two weeks as test data and the remaining data as the training set. The last 20% of the data in the training set constituted the validation set.

*5.1. Evaluation Metric.* Root mean square error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE) were used to evaluate the prediction performance of the model in each region. They are defined as

$$\begin{aligned}
\text{RMSE} &= \sqrt{\frac{1}{N}\sum_{n=1}^{\xi}\left(\widehat{D}_{i,n}^{p} - D_{i,n}^{p}\right)^2}, \\
\text{MAE} &= \frac{1}{N}\sum_{n=1}^{\xi}\left|\widehat{D}_{i,n}^{p} - D_{i,n}^{p}\right|, \\
\text{MAPE} &= \frac{100\%}{N}\sum_{n=1}^{\xi}\frac{\left|\widehat{D}_{i,n}^{p} - D_{i,n}^{p}\right|}{D_{i,n}^{p}},
\end{aligned} \tag{15}$$

where $\widehat{D}_{i,n}^{p}$ and $D_{i,n}^{p}$ are the predicted and real data, respectively, of the demand of region $i$ at time interval $j$ and $\xi$ is the number of test records.

*5.2. Parameter Settings.* The Pearson correlation coefficient was used to calculate the correlation between the pick-up demand data in the target region and the pick-up and drop-off demand data in other regions. Demand data with a strong linear correlation (absolute value of correlation coefficient greater than or equal to 0.7) were set as the exogenous data. $K_c$ (the number of layers of the CNN in the encoder) was set as 3. $q$ (the size of the convolution kernel matrix) was set as 5. $F_u$ (the number of image channels corresponding to each convolutional layer) was set as 64. $K_r$ (the number of layers of RHN in both the encoder and decoder) was set as 3. $l$ (the dimension of the RHN's hidden state in the encoder) was set as 128, as was $p$ (the dimension of the RHN's hidden state in the decoder). $L_c$, $L_p$, and $L_t$ (the length of the input data corresponding to different temporal properties) were set as 4, 2, and 2, respectively.

Table 1: Selected high-demand subregions.

| Identification number | Description |
|---|---|
| 48 | Clinton East |
| 79 | East Village |
| 142 | Lincoln Square East |
| 161 | Midtown Center |
| 162 | Midtown East |
| 163 | Midtown North |
| 170 | Murray Hill |
| 186 | Penn Station/Madison Square West |
| 230 | Time Square/Theatre District |
| 234 | Union Square |
| 236 | Upper East Side North |
| 237 | Upper East Side South |

*5.3. Methods for Comparison*

(1) Historical average (HA): this uses the average value of the previous demand at the positions given in the training set in the same relative time interval (i.e., the same time of day) to predict demand.

(2) Autoregressive integrated moving average (ARIMA): a classic model in time series prediction, it combines a moving average and autoregressive components to model time series. The ARIMA model needs to determine three parameters (P, I, and Q). In this experiment, we chose to call the pyramid library to automatically determine the relevant parameters.

(3) Linear regression (LR): LR uses the least square loss function of the linear regression equation to model the relationship between one or more of each of the independent and dependent variables. We used the Ridge and Lasso [32] linear regression models, and the tuning parameter of these models was set to 0.01.

(4) Multilayer perception (MLP): also known as an artificial neural network, the MLP has several hidden layers in addition to input and output layers. We used three hidden layers, each with 32 neurons.

(5) Extreme gradient boosting (XGBoost) [33]: XGBoost is a powerful boosting tree-based algorithm that is widely used in data mining. We set the learning rate to 0.1, and the remaining parameters took the default values.

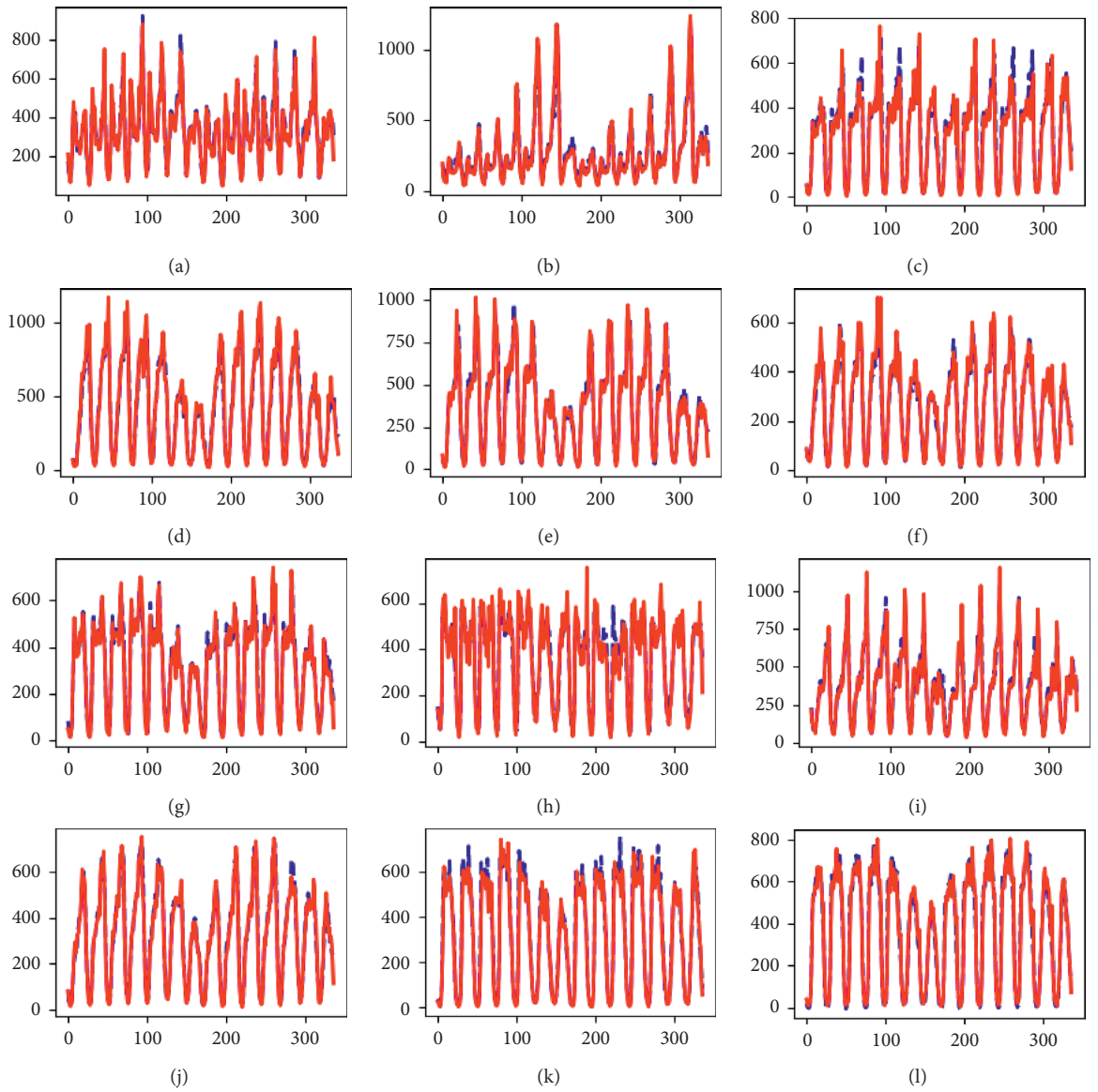


Figure 4: Results of MTR-HRHN prediction in different regions.

Table 2: Comparison with different baselines.

| Method | RMSE | MAPE (%) | MAE |
|---|---|---|---|
| HA | 116.89 | 38.9 | 81.90 |
| ARIMA | 76.24 | 39.4 | 55.92 |
| Ridge | 79.69 | 40.8 | 57.46 |
| Lasso | 79.75 | 41.0 | 57.51 |
| MLP | 66.02 | 21.5 | 45.59 |
| XGBoost | 65.63 | 18.0 | 44.28 |
| LSTM | 74.05 | 34.5 | 53.44 |
| Temporal view + spatial (neighbors) view | 60.13 | 19.5 | 42.49 |
| MTR-HRHN | 44.59 | 13.8 | 31.29 |

(6) Long short-term memory (LSTM) [17]: this method can deal with the problem of RNN gradient dissipation and has excellent performance in time series data processing. We selected a three-layer unidirectional LSTM network with 32 hidden layer nodes in each of the three layers.

(7) Temporal view + spatial (neighbors) view [10]: this spatiotemporal deep network uses CNN to extract spatially relevant information of the target region and its neighbor regions (those directly connected to the target region). The LSTM network processes the CNN output information to further extract temporal properties.

Table 3: Comparison of MTR-HRN with different time resolutions.

| Method | RMSE | MAPE (%) | MAE |
|---|---|---|---|
| HRHN_One | 50.47 | 13.8 | 33.95 |
| HRHN_Two | 44.62 | 13.8 | 30.94 |
| MTR-HRHN | 44.59 | 13.8 | 31.29 |

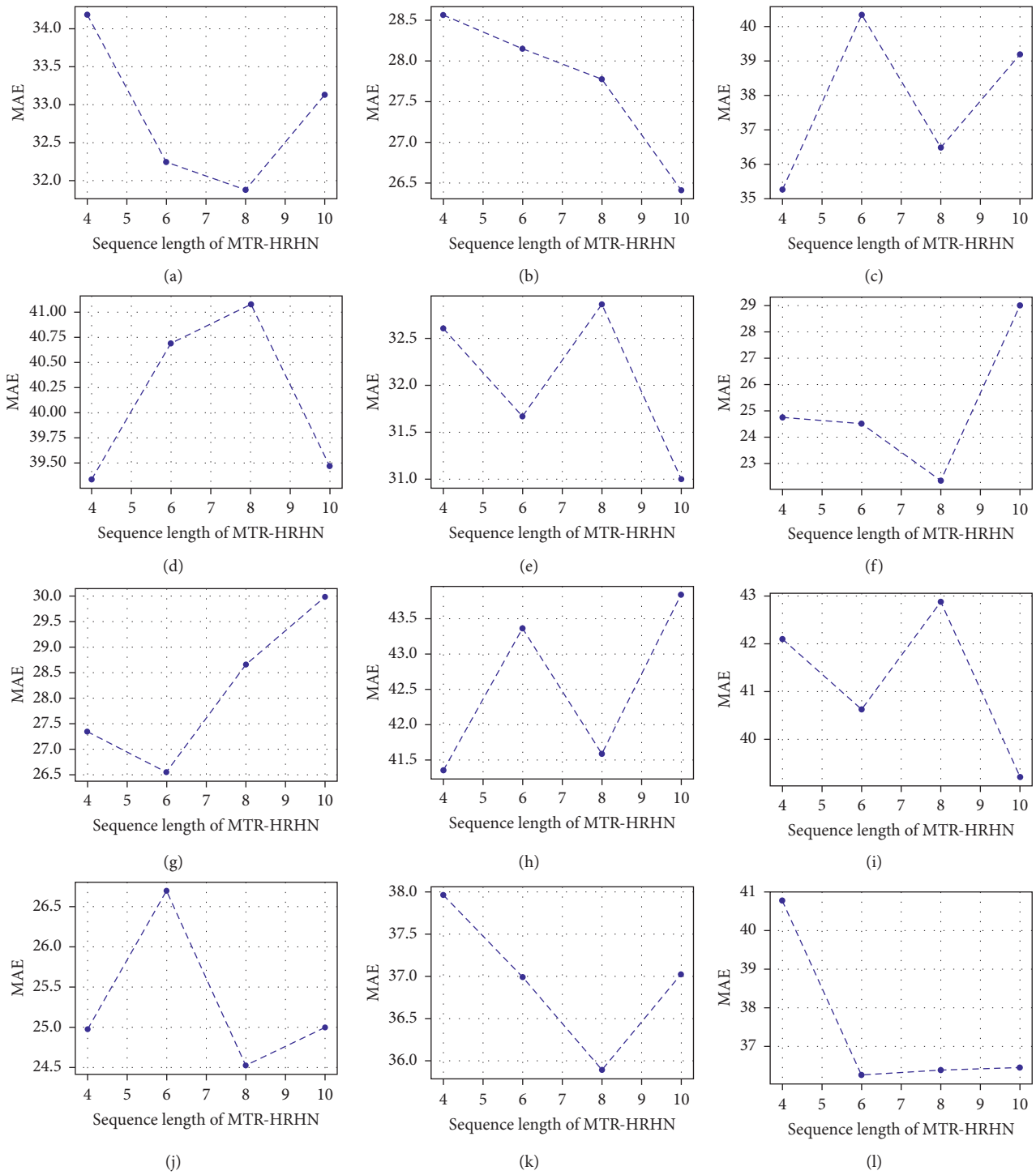


Figure 5: Experiments with different lengths of the input sequence.

We compared the prediction performance under single and multiple time resolutions of the following two models with that of the MTR-HRHN.

(1) HRHN_One: it only has one HRHN that models the closeness property of demand data.

(2) HRHN_Two: it has two HRHNs that model the closeness and period properties of demand data.

*5.4. Experimental Results and Analysis.* Figure 4 shows the fitting results of the predicted values of MTR-HRHN in the test set of the 12 high-demand regions of Manhattan, New York City. It can be found that the predicted results of MTR-HRHN are relatively accurate at most times. However, at the peak of each day, the deviation from the actual value is relatively large. This may be because the demand data in the peak times are more susceptible to nonnumeric data (such as

sudden bad weather or a social event), and MTR-HRHN does not put such data into the analysis.

Table 2 summarizes the results of all the methods. Compared to HA, MTR-HRHN reduces RMSE, MAPE, and MAE by 61.85%, 64.45%, and 61.8%, respectively. Compared to other nondeep learning models, MTR-HRHN reduces the RMSE, MAPE, and MAE by 43.37%, 65.84%, and 45.13%, respectively. MTR-HRHN also performs better than other deep learning models. Compared to LSTM, MTR-HRHN reduces RMSE, MAPE, and MAE by 39.78%, 59.98%, and 41.44%, respectively, and it reduces them by 25.84%, 29.05%, and 26.36% compared to the combination of CNN and LSTM.

The proposed MTR-HRHN model can generally obtain more accurate prediction results than the other models mentioned above. Compared to nonlinear models, MTR-HRHN can not only capture the dynamic connection of sequences in time but also extract spatial information. Compared to other deep learning models, MTR-HRHN can further extract the connections between different components of exogenous data at the same time and can expand the observable time pattern by introducing multiple time resolutions, thereby further enhancing the prediction performance.

Table 3 shows the experimental results of the time resolution test, from which it can be found that, compared to HRHN_One, HRHN_Two and MTR-HRHN have decreased errors (11.59%, 0%, and 8.86% and 11.65%, 0%, and 7.83%) in RMSE, MAPE, and MAE, respectively. Furthermore, choosing two time resolutions (corresponding to HRHN_Two) can greatly improve prediction accuracy. According to the comparison results of HRHN_Two and MTR-HRHN, the results in RMSE, MAPE, and MAE are almost equal. We can infer that it does not always improve the accuracy of prediction simply through using more time resolutions.

*5.5. Influence of Sequence Length.* Figure 5 shows the relationship between the future demand forecast performance of 12 regions and the length of the input sequence, from which it can be found that the forecast performance and length of the input sequence are not proportional. In general, the prediction performance first increases with the length of the input sequence. The model achieves locally optimal prediction performance when the sequence length reaches a certain value and begins to decline as the sequence length continues to increase. This is because RHN is essentially an extended LSTM network, and it faces the same disadvantage as RNN. So, when the sequence length is too short, the dynamic correlation information in time is not completely learned, and when the sequence length is too long, the difficulty of training convergence increases because many more parameters must be learned.

## 6. Conclusions

We applied the MTR-HRHN model to regional taxi demand prediction. By considering that real-world demand series typically exhibit patterns across multidimensional temporal patterns, MTR-HRHN employed three HRHNs to hierarchically extract and select the most relevant input features. It can capture the close, periodic, and trend characteristics of time series data. The experimental results show that the MTR-HRHN model achieves more accurate prediction results on demand data prediction than traditional time series prediction methods, classic machine learning regression models, and other deep learning models. We further compared and analyzed the impacts of the number of HRHN networks and the length of the input sequence on the prediction. These new factors shall be considered when applying the HRHN model or other spatiotemporal deep learning models to predict time series-related demands.

In subsequent research, we will optimize our model in two aspects. First, we will cluster the regions with the same demand patterns into one large region and use nonlinear correlation coefficient methods (such as a maximal information coefficient) to calculate the degree of correlation between the predicted region and other regions. Thus the strong correlation of exogenous sequences from the demand series of other regions can be captured. Second, many studies have shown that contextual data help to improve the prediction. We will collect some nonnumeric attributes (such as weather) and some point-of-interest information (such as functionalities of areas) and combine them with the historical exogenous data and/or historical target data. The new formatted input and its effort on the prediction will be further analyzed.

## Data Availability

The dataset selected for the experiment was the New York City Yellow Taxi Trip Records. The website is https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## References

[1] X. Zhan, X. Qian, and S. V. Ukkusuri, "A graph-based approach to measuring the efficiency of an urban taxi service system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 9, pp. 2479–2489, 2016.

[2] L. Zhang, T. Hu, Y. Min et al., "A taxi order dispatch model based on combinatorial optimization," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'17)*, pp. 2151–2159, Association for Computing Machinery, New York, NY, USA, 2017.

[3] H. Yang, Y. W. Lau, S. C. Wong, and H. K. Lo, "A macroscopic taxi model for passenger demand, taxi utilization and level of services," *Transportation*, vol. 27, no. 3, pp. 317–340, 2000.

[4] X. Li, G. Pan, Z. Wu et al., "Prediction of urban human mobility using large-scale taxi traces and its applications," *Frontiers of Computer Science*, vol. 6, no. 1, pp. 111–121, 2012.

[5] L. Moreira-Matias, J. Gama, M. Ferreira, J. Mendes-Moreira, and L. Damas, "Predicting taxi-passenger demand using

streaming data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1393–1402, 2013.

[6] Y. Li, J. Lu, L. Zhang, and Y. Zhao, "Taxi booking mobile app order demand prediction based on short-term traffic forecasting," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2634, no. 1, pp. 57–68, 2017.

[7] Y. Tong, Y. Chen, Z. Zhou et al., "The simpler the better: a unified approach to predicting original taxi demands based on large-scale online platforms," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '17)*, pp. 1653–1622, Association for Computing Machinery, New York, NY, USA, August 2017.

[8] D. Deng, C. Shahabi, U. Demiryurek et al., "Latent space model for road networks to predict time-varying traffic," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*, pp. 1525–1534, Association for Computing Machinery, New York, NY, USA, August 2017.

[9] Y. Tao, L. Ma, W. Zhang et al., "Hierarchical attention-based recurrent highway networks for time series prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 9, pp. 2479–2489, 2016.

[10] H. Yao, F. Wu, J. Ke et al., "Deep multi-view spatial-temporal network for taxi demand prediction," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, AAAI Press, New Orleans, LA, USA, pp. 2588–2595, February 2018.

[11] K. Cho, B. Van Merriënboer, D. Bahdanau et al., "On the properties of neural machine translation: encoder-decoder approaches," 2014, http://arxiv.org/abs/1409.1259.

[12] K. Cho, B. Van Merriënboer, D. Bahdanau et al., "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014, http://arxiv.org/abs/1406.1078.

[13] Y. Qin, D. Song, H. Chen et al., "A dual-stage attention-based recurrent neural network for time series prediction," 2017, http://arxiv.org/abs/1704.02971.

[14] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang, "Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction," *Sensors*, vol. 17, no. 4, p. 818, 2017.

[15] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI Press, San Francisco, CA, USA, pp. 1655–1661, February 2017.

[16] J. Zhang, Y. Zheng, D. Qi et al., "DNN-based prediction model for spatio-temporal data," in *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPACIAL '16)*, pp. 1–4, Association for Computing Machinery, New York, NY, USA, 2016.

[17] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[18] J. Chung, C. Gulcehre, K. Cho et al., "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, http://arxiv.org/abs/1412.3555.

[19] P. Bashivan, I. Rish, M. Yeasin et al., "Learning representations from EEG with deep recurrent-convolutional neural networks," 2015, http://arxiv.org/abs/1511.06448.

[20] S. C. Prasad and P. Prasad, "Deep recurrent neural networks for time series prediction," 2014, http://arxiv.org/abs/1407.5949.

[21] R. Yu, Y. Li, C. Shahabi et al., "Deep learning: a generic approach for extreme condition traffic forecasting," in *Proceedings of the 2017 SIAM international Conference on Data Mining*, pp. 777–785, SIAM, Houston, TX, USA, July 2017.

[22] J. Xu, R. Rahmatizadeh, L. Bölöni et al., "A sequence learning model with recurrent neural networks for taxi demand prediction," in *IEEE 42nd Conference on Local Computer Networks (LCN)*, pp. 261–268, IEEE, Singapore, October 2017.

[23] J. Xu, R. Rahmatizadeh, L. Bölöni et al., "Real-time prediction of taxi demand using recurrent neural networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 8, pp. 2572–2581, 2017.

[24] X. Zhou, Y. Shen, Y. Zhu et al., "Predicting multi-step citywide passenger demands using attention-based neural networks," in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pp. 736–744, Association for Computing Machinery, New York, NY, USA, 2018.

[25] D. Wang, W. Cao, J. Li et al., "DeepSD: supply-demand prediction for online car-hailing services using deep neural networks," in *IEEE 33rd international conference on data engineering (ICDE)*, pp. 243–254, IEEE, San Diego, CA, USA, April 2017.

[26] F. Rodrigues, I. Markou, and F. C. Pereira, "Combining time-series and textual data for taxi demand prediction in event areas: a deep learning approach," *Information Fusion*, vol. 49, no. 1, pp. 120–129, 2019.

[27] L. Liu, Z. Qiu, G. Li, Q. Wang, W. Ouyang, and L. Lin, "Contextualized spatial-temporal network for taxi origin-destination demand prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 10, pp. 3875–3887, 2019.

[28] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Training very deep networks," in *Proceedings of the 28th International Conference on Neural Information Processing Systems – Volume 2 (NIPS'15)*, MIT Press, Cambridge, MA, USA, pp. 2377–2385, 2015.

[29] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, http://arxiv.org/abs/1409.0473.

[30] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," 2014, http://arxiv.org/abs/1412.6980.

[31] X. Qian, S. V. Ukkusuri, C. Yang et al., "Forecasting short-term taxi demand using boosting-GCRF," in *Proceedings of the 6th International Workshop on Urban Computing ACM Transactions on Intelligent Systems and Technology (UrbComp 2017)*, pp. 53–61, Halifax, Canada, 2017.

[32] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 2017.

[33] T. Chen and C. Guestrin, "XGBoost: a scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*, pp. 785–794, Association for Computing Machinery, New York, NY, USA, August 2016.