A novel deep recurrent neural network for Short-term travel demand forecasting under on-demand ride services

Wei Wang1 and Xuedong Yan1*

1MOT Key Laboratory of Transport Industry of Big Date Application Technologies for Comprehensive Transport, Beijing Jiaotong University, Beijing, 100044, China
*Corresponding author’s e-mail: xdyan@bjtu.edu.cn

Abstract. Short-term travel demand forecasting throughout a city is crucial for passengers, drivers and the on-demand ride service platform, which could reduce waiting time and fuel consumption. In this paper, we propose a novel stacked bidirectional long short-term memory neural network (SBi-LSTMs) that can forecast short-term travel demand in each area of a city based on historical demand data and other relevant information. The proposed model is evaluated on the real-world data provided by China’s largest on-demand ride platform (DiDi Chuxing). The experimental results show that the SBi-LSTM outperforms other benchmark algorithms in predicting large-scale travel demand, such as ANN, RNN and LSTM. In addition, we analyzed the effects of different parameters on performance and training time.

1. Introduction

With the rapid development of mobile internet, especially the widespread popularity of smart phones, increasing people choose on-demand ride services. Although the matching rate of drivers and passengers has increased, it is difficult to ensure the uniformity of passenger travel demand and driver supply in temporal and spatial distribution due to the asymmetry of information and the randomness of demand on-demand ride services0. For the passengers, this mismatch leads to long wait times and decreased satisfaction. For the drivers, the search time is too long and the fuel consumption is increased. These will reduce the overall efficiency and profit of the on-demand ride service platform, so how to accurately predict short-term travel demand has become a top priority. It can help the platform to implement scheduling strategies in a timely manner, such as the incentive mechanism, which attracts drivers from oversupply to nearby overfull demand areas, improving passenger satisfaction, reducing time costs and energy consumption.

Large-scale urban network short-term demand forecasting is challenging due to a variety of potential factors. From the time dimension, although the passenger travel demand has a certain periodicity, each trip has a strong randomness, it is very difficult to accurately predict the number of trips in a short time interval; From the spatial perspective, the travel demand of each area is not only related to the attributes of the area, but also the attributes of its neighboring areas.

Although there are not many researches on the short-term travel demand forecasting problem under on-demand ride services, many studies on the predictions of other traffic problems such as speed and travel flow are worth learning, which are mainly based on classical statistical methods or deep learning approaches[2].

Traditional statistical models such as Kalman filtering and ARIMA play an important role in the prediction of traffic problems. Zhang et al. (2011) presented a Kalman filter method to forecast short-term passenger flow based on the characteristic analysis[3]. Moreira-Matias et al. (2013) predicted the
spatial distribution of taxi-passengers for a short-term time horizon utilizing three time-series forecasting techniques that combined time-varying passion model and ARIMA[3]. Chen et al. (2016) also predict the passenger flow data based on the historical data utilizing the ARIMA model which parameters were identified by the autocorrelation and partial autocorrelation function[4]. Zhao et al. (2016) implemented compared three predictors: the Markov predictor, the Lempel-Ziv-Welch predictor, and the Neural Network predictor to predict taxi demand at high spatial resolution[5].

With the explosive growth of data and the improvement of computer power in recent years, more and more deep learning methods are applied to traffic prediction problems[6]. Tsai et al. (2009) proposed multiple temporal units neural network (MTUNN) and parallel ensemble neural network (PENN) to forecast short-term railway passenger demand[7]. Huang et al. (2014) proposed a deep architecture that consists of two parts, i.e., a deep belief network (DBN) at the bottom and a multitask regression layer at the top to predict short-term traffic flow[8]. Ma et al. (2015b) attempts extended a deep Restricted Boltzmann Machine and Recurrent Neural Network architecture for the large-scale traffic network analysis to predict the evolution of traffic congestion with the help of taxi GPS data[9].

Since the short-term forecasting of traffic demand is a prediction problem for time series, the family of recurrent neural networks (RNN)[10] that can capture the characteristics of time series data was widely used. The input of each time step is not only the input of the current time step, but also the information left over from the previous time step in RNNs. Brébisson et al. (2015) used bidirectional recurrent neural networks to predict a fixed-length output from a variable-length sequence[11]. But the traditional RNN architecture lacks the ability for long-term storage [12]. In order to overcome the shortcoming, Long Short-Term Memory (LSTM), a special type of RNNs, was presented which can capture long-term dependencies due to the gating mechanism [13]. Cheng et al. (2016) compared three models, including the deep neural network, the stacked long short-term memory (LSTM) network, and the feature-level data fusion model to forecast the day-today travel demand[14]. LSTM has been applied to many other fields such as speech recognition [15], visual recognition [16], language modeling [17].

The traditional LSTM structure only utilizes information before the predicted time interval [18]. However, the travel demand has a strong periodicity, and the information after the predicted time interval also needs to be learned. M. Schuster et al. [19] proposed a bidirectional LSTM with the ability to capture both forward and backward information which are proven especially useful for works like sentiment and phoneme classification [20] and speech recognition [21].

In this paper, we propose a novel SBi-LSTM model which is made up of multiple Bi-LSTMs layers stacked on each other, to deal with historical demand data and other spatio-temporal variables in the short-term travel demand forecasting for the on-demand ride service platform. Different from the aforementioned studies, we encode and transform the data into time series for large-scale networks before feeding these variables into the structure. With this formulation, our model can simultaneously predict travel demand across all regions of the city, rather than simply predicting travel demand for one particular region. Validated by the real-world data provided by China's largest on-demand ride platform, our model outperforms other benchmark algorithms in predicting large-scale travel demand, such as ANN, RNN and LSTM.

The rest of this paper is organized as follows. Section 2 first describes data encoding and network-wide time series data, and then introduces the structure and mathematical formulation of the proposed SBi-LSTM. Section 3 evaluates the proposed model on a dataset compare our model with other benchmark algorithms. Finally, Section 4 concludes the paper and outlooks future research.

2. Methodology.

2.1. Data cleaning and encoding
The floating car datasets utilized in this paper are extracted from the on-demand ride service platform. Each of order consists of order ID, requesting time, travel distance, ending time, longitude and latitude of the origin and destination. There will be some errors in the process of collecting, transmitting and recording the data, and some positioning data errors will also be generated in the GPS. These errors will
greatly reduce the accuracy of the data. Therefore, it is important to remove the wrong order information that is incompatible with the physical phenomena of traffic. The data was cleaned by the following steps.

Step1: Remove the data with missing attributes and the orders outside the scope of the study.

Step2: Excluding orders with an average travel speed which is out of the range 0 -120km/h.

Step3: Calculate the Euclidean distance between ODs and filter according to the ratio of actual driving distance to Euclidean distance in the orders. It is statistically found that the 5% quantile is about 1, and the rejection ratio is less than 5% quantile and greater than 95% quantile Order.

Since the demand is displayed as separate points on the map, in order to avoid the complexity of map matching in large-scale network demand forecasting, we divide the research area into several grids to predict short-term demand more quickly. For the sake of reproducibility, we use the Geohash system in this paper. Geohash is spatial data geocoding system that is a hierarchical spatial data structure which subdivides space into buckets of grid shape. It can encode two-dimensional spatial latitude and longitude data into a string using the 32 letters of 0-9, b-z (removing a, i, l, o). For example, the geographic coordinates of (116.390705, 39.923201) can be converted to 'wx4g0ec1', achieving a column index. Different coding precisions represent different grid sizes.

The research range selected in this paper is longitude: 116.19° E -116.54 ° E (30KM), latitude: 39.77 ° N-40.0341796875 ° N (29.3KM), within the 5th Ring-road Expressway in Beijing, China. It is divided into 1536(48*32) grids with a code length of 6. The side length of each grid is about 608m * 936m at 39-40 ° N.

### 2.2. Network-wide Time Series Data

Large-scale urban network short-term demand forecasting is a time series forecasting problem that requires the conversion of historical data into usable time-series data. Since the demand in each region is not only related to the historical demand of the region and the surrounding areas, but also the hour, date and weather of the predicted time interval (Ke et al., 1999). To take these network-wide influences into account, we aggregate the various types of data in $M$ grids at 10 minutes intervals in this paper.

We have selected the following parameters:

- Demand at the $t$th time interval of the $m$th grid is defined as the number of orders during the time interval within the grid, which is denoted by $D_t^m$.

- Hour of day at the $t$th time interval is defined as the number of hours, which is denoted by $H_t$.

- Due to some short holidays (e.g. Qing Ming Festival, Dragon Boat Festival, Spring Festival, Labor Day) which seriously affect travel behavior in China, we also denote another dummy variable $DT_t$ to be the date status, which catches up the distinguished properties between holidays, weekdays and weekends.

- By analyzing the data set, it is found that the demand distribution has obvious regularity with time. Thus we introduced the hours and dates as variables. According to the difference between hour of day and date status, historical average demand, maximum demand, minimum demand, and median demand at the $t$th time interval are denoted by $D_t^{mean}$, $D_t^{max}$, $D_t^{min}$, $D_t^{med}$.

- We consider 2 categories of weather variables, including temperature and rainfall capacity, which include 5 different variables. Depending on various of hour of day and date status, the average temperature, maximum temperature, minimum temperature of the day at the $t$th time interval are denoted by $T_{t}^{mean}$, $T_{t}^{max}$, $T_{t}^{min}$. Rainfall capacity (measured per hour) at the $t$th time is denoted by $R_t$.

Input to our experiment is a matrix (i.e. $X_T^M$) which contains all variables of the $n$ historical time intervals (steps). The target of our experiment is to predict the demand of each grid in the next time interval (i.e. $Y_T^M$).

$$X_T^M = \begin{bmatrix}
D_{T-1}^1 & D_{T-1}^2 & \ldots & D_{T-1}^M \\
D_{T-1}^2 & D_{T-1}^2 & \ldots & D_{T-1}^M \\
\vdots & \vdots & \ddots & \vdots \\
D_{T-1}^M & D_{T-1}^M & \ldots & D_{T-1}^M \\
\end{bmatrix} \begin{bmatrix}
D_{T-a}^1 & D_{T-a}^2 & \ldots & H_{T-a} & D_{T-a}^{max} & D_{T-a}^{min} \\
D_{T-a}^2 & D_{T-a}^2 & \ldots & H_{T-a} & D_{T-a}^{max} & D_{T-a}^{min} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
D_{T-a}^M & D_{T-a}^M & \ldots & H_{T-a} & D_{T-a}^{max} & D_{T-a}^{min} \\
\end{bmatrix} \begin{bmatrix}
D_{T-a}^{max} & D_{T-a}^{max} & \ldots & T_{T-a}^{max} & H_{T-a} & D_{T-a}^{max} \\
D_{T-a}^{max} & D_{T-a}^{max} & \ldots & T_{T-a}^{max} & H_{T-a} & D_{T-a}^{max} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
D_{T-a}^{max} & D_{T-a}^{max} & \ldots & T_{T-a}^{max} & H_{T-a} & D_{T-a}^{max} \\
\end{bmatrix} \begin{bmatrix}
D_{T-a}^{max} & D_{T-a}^{max} & \ldots & T_{T-a}^{max} & R_{T-a} \\
D_{T-a}^{max} & D_{T-a}^{max} & \ldots & T_{T-a}^{max} & R_{T-a} \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
D_{T-a}^{max} & D_{T-a}^{max} & \ldots & T_{T-a}^{max} & R_{T-a} \\
\end{bmatrix}$$
\[ Y^M_f = [D^0_f, D^1_f, \ldots, D^{m-1}_f] \] (2)

The One-Hot-Encoder is used here to process the dummy variables and convert the eigenvalues of the variables into sparse matrices. To avoid the influence of the various data units on the result, Min-Max-Scaler (see in Eq. 3) is used to normalize the data, so that each column in the matrix \( X^M_f \) is converted into a dimensionless index, that is, each index value at the same quantitative level.

\[ Z = \frac{Z - Z_{\text{min}}}{Z_{\text{max}} - Z_{\text{min}}} \] (3)

2.3. LSTM Sequence Learning Model

Long Short-Term Memory (LSTM), a special type of RNNs, can capture long-term dependencies due to the gating mechanism [22]. There are also LSTM cells in the hidden layer of LSTM which is shown in Fig. 1. Each LSTM cell contains input gate, output gate, forget gate and cell state, denoted as \( i_t, o_t, f_t \), and \( c_t \) at the \( t \)th time interval respectively which can be calculated by the following equations:

\[ i_t = \sigma (W_{xi}x_t + W_{hi}h_{t-1} + b_i) \] (4)
\[ f_t = \sigma (W_{xf}x_t + W_{hf}h_{t-1} + b_f) \] (5)
\[ c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \] (6)
\[ o_t = \sigma (W_{xo}x_t + W_{ho}h_{t-1} + b_o) \] (7)
\[ h_t = o_t \odot \tanh(c_t) \] (8)

Where \( x_t \) is the input matrix at the \( t \)th time interval. \( W_{xi}, W_{hi}, W_{xf}, W_{hf}, W_{xc}, W_{hc}, W_{xo}, W_{ho} \) are the weighted matrices for transformation from the first subscript to the second subscript. \( b_i, b_f, b_c, b_o \) denote bias vectors. \( \odot \) represents element-wise multiplication with the same dimensions. \( \sigma \) is the activation (logistic sigmoid) function, and \( \tanh \) is the hyperbolic tangent function given by following equations:

\[ \sigma_x = \frac{1}{1 + e^{-x}} \] (9)
\[ \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \] (10)

Bi-LSTMs, based on bidirectional RNN, can process historical and future data in both directions. Bi-LSTMs have separate two hidden layers which connect to the same output layer, as shown in Fig. 2. Each of time interval has an output calculated by the Eq. 11, where \( f \) is a concatenating function.

\[ y_t = f(h_t, \tilde{h}_t) \] (11)
The SBi-LSTM model is made up of multiple Bi-LSTMs layers stacked on each other which can deepen the complexity of the network to explain more underlying relationships. The object is to minimize the mean squared error between the estimated and real demand value, through which the weighted and bias parameters can be trained in our experiment (see in Eq. 12). The first term of the objective function means the mean squared error, and the second term represents L2-norm regularization. $\lambda$ is the regularization parameter which balances between training data well and keeping parameter values suitable. $\theta$ stands for all parameters in training.

$$\text{loss} = \min_{W, \theta} \left( \| x - \tilde{x} \|^2_2 + \lambda \| \theta \|^2_2 \right)$$  \hspace{1cm} (12)

3. Experiment and results

3.1. Model comparisons

In this section, we test and compare the proposed SBi-LSTM model with other baselines utilizing the car orders and the weather data. The car order dataset is extracted from DiDi, the largest on-demand ride service platform offering ride-hailing services for 20 million registered taxi drivers in 380 cities in China, during half of a year between May 21, 2017 and October 31, 2017 (163 days). We obtained the full sample data in the study area which includes around 78 million taxi trips after data filtering. Then we count the number of taxi requests every 10 minutes (time interval). In such a way, historical taxi data in each area becomes the number of car requests data sequences. Combined with weather data, we normalize all the datasets to the range $[0, 1]$, which sequences are fed into the model for sequential patterns learning. In summary, the dataset has $6 \times 24 \times 163$ time intervals in total. Suppose we set the time step to 7 which means the model uses 70 minutes consecutive time series data to predict the following 5-minute travel demand value, the dataset is reshaped and the total number of sequences is $N=23465$ (23472–7). The features consisting of the number of grids, impacting factors such as hour of day, date status, weather conditions and other historic information in each time interval. In conclusion, the encoded input data shape is $(23465, 7, 1571)$.

To avoid using future information, the dataset is divided into 60% training dataset consisting of observations between May 21, 2017 and August 26, 2017 (98 days), and the 40% test dataset consisting of the rest between August 26, 2017 and October 31, 2017 (65 days). Fig. 3 shows the total demand for all grids over different dates, the date of training set is before the red dashed line, and the date of test set is after the red dashed line. It can be observed that the training set and testing set are sometimes abnormally low. The first low-demand period is due to the Chinese Ching Ming holiday (in the training set), while the second one is affected by the National Day holiday (in the test set). Especially in giant cities such as Beijing, the demand for these abnormal periods increases the difficulty of short-term demand forecasting.

Fig. 4 shows travel demand distribution characteristics within one day according to different date status types. It can be seen that the travel demand on working days is the highest, and there are three peaks in one day; the total amount on the weekend is slightly smaller, but only shows the characteristics
of a single peak; the total travel demand for the holiday is the smallest, and the trend is similar to the weekend.

![Figure 3. Total demand for all grids over different dates](image3.png)

![Figure 4. Demand distribution within one day according to different date status types](image4.png)

To measure the performance of the SBi-LSTM model, we must choose some basic models for comparison. Since various types of neural networks perform much better than traditional methods such as ARIMA and Kalman filtering in time series prediction, we only compare the performance of ANN, RNN, LSTM, GRU and the SBi-LSTM model.

- **ANN**: The artificial neural network here means the simplest fully connected feed-forward neural network which is commonly used for classification and regression problems. Correspondingly, the input data will be converted from 3-D to 2-D.

- **RNN, LSTM, GRU**: The networks can store historic information with the special mechanism to deal with sequence data. The inputs of these models are the same as SBi-LSTM model. But none of the models are stacked.

To examine the performance of our SBi-LSTM model, the effectiveness of different algorithms are systematically compared via root mean squared error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE), epochs, and training time (t) using the following equations:
\[ RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2} \]  

\[ MAE = \frac{1}{N} \sum_{i=1}^{N} |y_i - \hat{y}_i| \]  

\[ MAPE = \frac{1}{N} \sum_{i=1}^{N} \frac{|y_i - \hat{y}_i|}{y_i + \hat{y}_i + c} \]

Where \( N \) is the number of grids, \( y_i \) and \( \hat{y}_i \) are the \( i \)th grid truth and the estimated value of the demand, respectively. The purpose of setting the constant \( c \) (equal to 1 in this study) in Eq. 15 to avoid division by zero when actual and predicted values are both 0.

Table 1 shows the predictive performance of the SBi-LSTM model and other benchmark models. It can be found that the simplest ANN performs worst which just need around two minutes to train. The performance of RNN, LSTM and GRU have not many differences, due to they have a high degree of structural similarity. The SBi-LSTM has the best performance in almost all indicators which achieves the best predictive performance with the RMSE of 1.823, MAE of 1.027 and MAPE of 0.196. Although the training time of SBi-LSTM is much longer than other models, it is still within the acceptable range, and the predicted time in the actual application is less than 1 second. It indicates that bi-direction is valuable to the model since it takes into account historical and future data in both directions.

Fig. 5 shows a density map of real and predicted travel demand over the entire city at 18:00 during evening peak hours, where red grids represent the area with the highest demand, the orange grid represents the area with relatively low demand, and the green grid represents the area with the lowest demand. The figure shows the needs of different grids are significantly different across space. Grids with high demand are concentrated in work and business districts such as Wangjing, Wangfujing, Guomao and Zhongguancun, especially at the evening peak. Demand in the eastern and northern parts of Beijing is significantly higher than in the west and south, consistent with actual economic development. Fig. 6 shows the spatial and temporal distribution of the real and predicted travel demand from grid 750 to grid 800 where are located in the central business district (In the white box shown in fig. 5). The x-axis refers to 24 hours a day, while the y-axes represents different grid numbers. Red shows high demand during this time interval in this region. It is obvious that the travel demand during the day is much greater than the night. On weekdays, there are two peaks in the region, but there is only a short-lived peak after 18 o’clock on weekends. Even on the same day, different regions have different travel demands and trends, which makes it hard to forecast short-term passenger demand.

Overall, the density map illustrates that only a small difference between the predicted and real values, which suggests that the SBi-LSTM can capture the spatio-temporal characteristics of the travel demand to predict accurately. With less than one second of real-time forecasting and demand maps, we can clearly and quickly detect the state of demand distribution, adjust supply strategies in a timely manner, and prevent imbalances between supply and demand.

|         | RMSE  | MAE   | MAPE  | Epochs | Time(s) |
|---------|-------|-------|-------|--------|---------|
| ANN     | 1.903 | 1.065 | 0.201 | 55     | 110     |
| RNN     | 1.847 | 1.040 | 0.197 | 76     | 228     |
| LSTM    | 1.848 | 1.042 | 0.199 | 56     | 224     |
| GRU     | 1.865 | 1.046 | 0.200 | 66     | 264     |
| SBi-LSTM| 1.823 | 1.027 | 0.196 | 68     | 1496    |
3.2. Sensitivity analysis

In this section, we change several parameters to do sensitivity analysis based on the SBi-LSTM which includes time steps, layers of the structure and the look-ahead time windows (Multi-step time window prediction).

First, we train the SBi-LST based on different time steps with the three-layer structure. From Fig. 7(a), it is obvious that the training time increases linearly while RMSE first drops and then stabilizes with the increase of time step. This can be explained by the fact that as the input information increases, the training consumption time increases, and the predictive performance also becomes better. However, when historical information that has little influence on the current stage is input, the influence on the prediction accuracy becomes small, and even the prediction accuracy is affected. As we can see from Fig. 7(b), training time increases rapidly with the increase of layers, but RMSE increases slowly. This phenomenon could be explained as one layer of the network can already capture enough historical information to predict.
We chose two indicators of RMSE and MAE for performance comparison of the multi-step prediction which predicts different time intervals (t+n) using the same historical information. Fig. 7(c) demonstrates that both indicators become slightly larger as the prediction interval increase. The results show that the model is more accurate for near-time interval predictions, but also multi-step predictions.

Fig7. Sensitivity analysis of parameters

4. Conclusion and future work
We propose a novel stacked bidirectional long short-term memory neural network (SBi-LSTMs) for short-term travel demand forecasting. Using the spatial data geocoding (Geohash) system, we encode the multivariate data (e.g. historical travel demand, number of hours, the date status, weather conditions, historical average demand, maximum demand, minimum demand, and median demand) and input it into the model which is compared with several benchmark algorithms including the ANN, RNN and LSTM. The model is validated on the more than five months’ travel demand data which is full sample data from DiDi Chuxing in Beijing, China. Experimental results show that the SBi-LSTM outperforms other benchmark algorithms in predicting large-scale travel demand, which indicates that our model can capture the spatio-temporal characteristics of the travel demand more accurately.

Future work can be extended by adding other types of data into the model, such as points of interest and real-time traffic flow. In addition, travel demand forecasting is only the first step of traffic planning. Due to the lack of supply for each area of the urban network in the future, it is not well coordinated with the dispatch center of the on-demand ride service platform. The study of the difference between supply and demand should be investigated.

References
[1] Moreira-Matias, L., Gama, J., Ferreira, M., Mendes-Moreira, J., Damas, L., 2013. Predicting taxi-passenger demand using streaming data. IEEE Trans. Intell. Transp. Syst. 14 (3), 1393–1402.
[2] Q. Ye, W. Y. Szeto, and S. C. Wong, “Short-term traffic speed forecasting based on data recorded at irregular intervals,” IEEE Transactions on Intelligent Transportation Systems, vol. 13, no. 4, pp. 1727–1737, 2012.
[3] Zhang, C.H., Song, R., Sun, Y., 2011. Kalman filter-based short-term passenger flow forecasting on bus stop. J. Transport. Syst. Eng. Inform. Technol. 11 (4), 154–159.

[4] Chen, Y.L., Sha, Y.W., Zhu, X.L., Zhang, X.H., 2016. Prediction of shanghai metro line 16 passenger flow based on time series analysis—with Lingang avenue station as a study case. Operat. Res. Fuzziol. 6 (1), 15–26.

[5] Zhao, K., Khryashchev, D., Freire, J., Silva, C., Vo, H., 2016. Predicting taxi demand at high spatial resolution: approaching the limit of predictability. In: IEEE International Conference on Big Data.

[6] Vlahogianni E I, Karlaftis M G, Golas J C. Short-term traffic forecasting: Where we are and where we’re going[J]. Transportation Research Part C Emerging Technologies, 2014, 43(1).

[7] Tsai, T.H., Lee, C.K., Wei, C.H., 2009. Neural network based temporal feature models for short-term railway passenger demand forecasting. Expert Syst. Appl. 36 (2), 3728–3736.

[8] Huang, W., Song, G., Hong, H., Xie, K., 2014. Deep architecture for traffic flow prediction: deep belief networks with multitask learning. IEEE Trans. Intell. Transp. Syst. 15, 2191–2201.

[9] Ma, X., Yu, H., Wang, Y., Wang, Y., 2015b. Large-scale transportation network congestion evolution prediction using deep learning theory. PloS one 10, e0119044.

[10] Williams, R.J., Zipser, D., 1989. A learning algorithm for continually running fully recurrent neural networks. Neural Comput. 1, 270–280.

[11] A. de Brébisson, É. Simon, A. Auvolat, P. Vincent, and Y. Bengio. (2015). “Artificial neural networks applied to taxi destination prediction.”

[12] Sutskever, I., Hinton, G.E., Taylor, G.W., 2009. The recurrent temporal restricted boltzmann machine. In: Advances in Neural Information Processing Systems, pp. 601–1608.

[13] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[14] Cheng, Q., Liu, Y., Wei, W., Liu, Z., 2016. Analysis and forecasting of the day-to-day travel demand variations for large-scale transportation networks: a deep learning approach.

[15] Zhao, Jianfeng, Xia Mao, and Lijiang Chen. "Speech emotion recognition using deep 1D & 2D CNN LSTM networks." Biomedical Signal Processing and Control 47 (2019): 312-323.

[16] K. Simonyan and A. Zisserman. (2014). “Very deep convolutional networks for large-scale image recognition.”

[17] A. Karpathy, J. Johnson, and L. Fei-Fei. (2015). “Visualizing and understanding recurrent networks.”

[18] Z. C. Lipton, J. Berkowitz, and C. Elkan, “A critical review of recurrent neural networks for sequence learning,” arXiv preprint arXiv:1506.00019, 2015.

[19] M. Schuster and K. K. Paliwal, “Bidirectional recurrent neural networks,” IEEE Transactions on Signal Processing, vol. 45, no. 11, pp. 2673–2681, 1997.

[20] A. Graves and J. Schmidhuber, “ Framewise phoneme classification with bidirectional lstm and other neural network architectures,” Neural Networks, vol. 18, no. 5, pp. 602–610, 2005.

[21] A. Graves, N. Jaitly, and A.-r. Mohamed, “Hybrid speech recognition with deep bidirectional lstm,” in Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on. IEEE, 2013, pp. 273–278.

[22] K. Greff, R. K. Srivastava, J. Koutn’ik, B. R. Steunebrink, and J. Schmidhuber, “Lstm: A search space odyssey,” IEEE transactions on neural networks and learning systems, 2017.

[23] Chollet, F., et al., 2015. Keras. <https://github.com/fchollet/keras>.

[24] Nair, Vinod, and Geoffrey E. Hinton. "Rectified linear units improve 366 restricted boltzmann machines." Proceedings of the 27th international 367 conference on machine learning (ICML-10). 2010.
[25] Hinton G E, Srivastava N, Krizhevsky A, et al. Improving neural networks by preventing c=o-adaptation of feature detectors[J]. arXiv preprint arXiv:1207.0580, 2012.