Human Factors in Software and Systems Engineering, Vol. 94, 2023, 54–63
https://doi.org/10.54941/ahfe1003770

Short-Time Taxi Demand Prediction Based on Transformer-LSTM in Integrated Transportation Hub

Wenjuan Zhang[1], Xiujie Li[1], Bin Zhang[1], Haozhe Yang[1], and Guangbin Wang[2]

[1]School of Mechanical and Energy Engineering,Tongji University, Shanghai, China
[2]School of Economics and Management, Tongji University, Shanghai, China

ABSTRACT

Taxis is an important tool to collect and distribute passenger flows in large-scale transportation hubs. Accurately predict the demand for taxis is very helpful and necessary to improve the service level and efficiency of the hub and serve the dynamic decision-making of taxi drivers. Based on artificial intelligence deep learning method, this paper builds a short-term taxi demand forecasting model to match the taxi demand of passengers with the supply of taxis more reasonably. By fully mining time-series characteristics of taxis flow historical data, the model integrates the Transformer and the LSTM neural network,can short-term predict demand for taxis every 15 minutes. Taking Shanghai Hongqiao hub as an example, the experiment collected several months of taxi cross-section traffic data to train the model. The results shows that our trained Transformer-LSTM model has a high accuracy in predicting short-term taxi demand. In order to verify the superiority of the model, we compare the model with other mainstream CNN, LSTM and other baseline prediction models. The experimental results show that the comprehensive performance of our model has the highest accuracy. We hope this paper can provide a powerful reference for the optimization and improvement of taxi dispatching and overall operations in transportation hubs.

Keywords: Taxi demand prediction, Transformer, LSTM, Transportation hubs

INTRODUCTION

With the rapid development of transportation modes, passenger flow in large transportation hubs is increasing year by year, which poses challenge to hub's operational capacity. Inconvenient interchange methods affect passengers' satisfaction largely, and many passengers stranded in the hubs can also be a safety hazard (Zhong et al. 2020). As one of the main modes of transferring within the hub, taxis scheduling still have many problems, such as long transfer times and long queues of waiting. Therefore, basing data to forecast the demand for taxis in the hub is necessary. Traffic data contains characteristics of passenger flow, using intelligent algorithms to deep mine the data can provide powerful and efficient on passenger flow prediction.

Transportation related forecasting has been studied for a long time, such as airport passenger flow forecasting (Lin et al. 2022) and taxi demand

forecasting (Rodrigues et al. 2019). Traditional methods for traffic prediction include ARIMA models (Chen et al.2019) and Kalman filters(Jiao et al. 2016). With the development and application of traffic big data, more and more scholars' research moves to deep learning methods, such as the commonly used improved RNN, including LSTM (Han et al. 2019; Wang et al. 2020), GRU (Yang et al. 2019), and CNN in combination with other methods(Liu et al. 2020). These researches contribute great value to hub taxi demand forecasting, but most models ignore the importance of short-time forecasting and data periodicity features, which leads to insufficient accuracy and timeliness of the forecast. This paper tries to propose a combined deep learning short-time prediction model combining LSTM neural network and Transformer, to process time series data and fully exploiting the time periodicity features.

PROBLEM DESCRITIONS

This paper focuses on taxi demand forecasting in large transportation hubs. The paper takes the historical data of taxi departures from the storage yards in the past, and forecasts the departures in the future period. The mathematical description of the problem is as follows: at a given time step t, the sequence $H_{T_1}=\{y_{t-T_1-1}, y_{t-T_1}, .., y_t\}$of the previous T_1 steps of the historical data is used to predict the sequence $H_{T_2}=\{y_{t+1}, y_{t+2}, .., y_{t+T_2}\}$ corresponding to the next time step T_2. T_1 denotes the input step of the prediction model, T_2 denotes the output step of the prediction model, and y_i denotes the historical departure volume for time step i.

Suppose $D=\{y_1, y_2, .., y_n\}$ is denoted as a time series of length n time steps, then before model prediction, it is also necessary to reconstruct the sequence D in the form of $H_{T_1}, H_{T_2}$with a sliding window, T_1 for an input step of and an output step of T_2, the sliding window is T_1+T_2, i.e., for each sliding one unit, it produces n-T_1 sequences of length T_1+T_2, where the first T_1 data of each sequence are used as input X_i and the last T_2 data are used as output Y_i. After data reconstruction, the input set of the model is $X_{\text{in}}=\{X_1, X_2, \ldots, X_j\}$, and the output set is denoted as $Y_{\text{out}}=\{Y_1, Y_2, \ldots, Y_j\}$.

We focuses on the single-step prediction problem, i.e., $T_2 = 1$, using data from the past T_1 time steps to predict data from the next time step, so the pivotal taxi prediction problem defined in this paper can be expressed by the following equation:

$$Y_t = G((X_{t-T_1}, \ldots, X_{t-1})) \tag{1}$$

where, $G(\cdot)$ denotes the mapping function, Y_t is the model prediction at time step t, and $(X_{t-T_1}, \ldots, X_{t-1})$ denotes the sequence of the past T_1 time steps.

METHODS

This paper proposes a combined deep learning prediction model, namely the Transformer-LSTM model, which mainly contains two parts, the first part is LSTM neural network layer, and the second part is the improved Transformer encoder. The structure of the model is shown in Fig. 1.

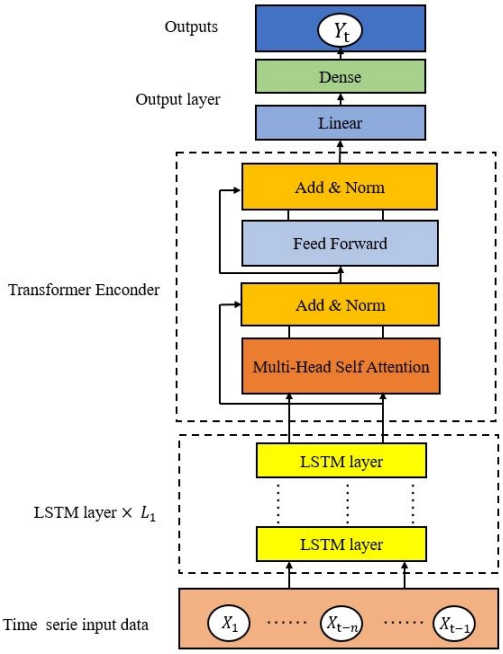


Figure 1: Model structure.

LSTM NEURAL NETWORK LAYERS

Due to the poor performance of RNN in long sequences, Hochreiter (Hochreiter et al. 1997) proposed LSTM neural network, which is a special type of RNN that introduces "cell" in the structure to record additional information. We use LSTM neural network to mine the continuity features of the input time series. The structure of the LSTM unit is shown in Fig. 2.

In Fig. 2, x_t denotes the input at time step t and h_t denotes the output at time step t. The formulas for the whole structure are as follows.

$$f_t = \sigma\left(W_f \cdot \left[h_{t-1}, x_t\right] + b_f\right) \tag{2}$$

$$i_t = \sigma\left(W_i \cdot \left[h_{t-1}, x_t\right] + b_i\right) \tag{3}$$

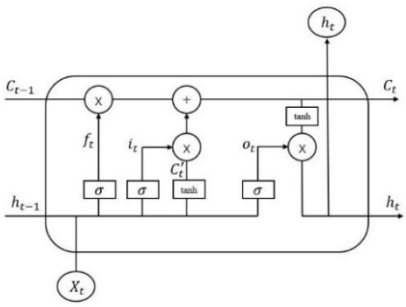


Figure 2: LSTM unit structure.

$$C_t^{'} = \tanh\left(W_C \cdot \left[h_{t-1}, x_t\right] + b_C\right) \quad (4)$$

$$C_t = f_t * C_{t-1} + i_t * C_t^{'} \quad (5)$$

$$o_t = \sigma\left(W_o \cdot \left[h_{t-1}, x_t\right] + b_o\right) \quad (6)$$

$$h_t = o_t * \tanh\left(C_t\right) \quad (7)$$

where $\sigma(\cdot)$ is the sigmoid activation function with the output range [0,1]; tanh$(\cdot)$ is the hyperbolic tangent activation function; W_f, W_i, W_C, W_o are the weight matrices; b_f, b_i, b_C, b_o are the bias terms.

The number of LSTM combinatorial model layer is L_1, the number of hidden layers by each LSTM layer is 64, and the output features are set to 10. Assuming an input step size of d, the LSTM layer outputs a (10 × d) matrix, which is represented as the result of learning continuity features from the LSTM layer.

TRANSFORMER LAYERS

In the prediction of time series, the input and output are continuous, and there is no semantic position correspondence, so this paper discards the position encoding method and directly concatenate with the LSTM layer mentioned above to make up for the shortcomings of transformer in extracting temporal features. In this layer, the encoder part of the original Transformer model is retained, and the encoder includes the feed-forward network layer and the self-Attention layer; the model discards the mask operation of the encoder and uses the multi-headed self-attention mechanism to mine the temporal features, and finally outputs the results through the linear layer.

MULTI-HEAD SELF-ATTENTION MECHANISM

Vaswani (Vaswani et al. 2017) proposed the Transformer model in 2017, which uses the self-attention structure instead of the RNN structure commonly used in NLP tasks, and its biggest advantage over the RNN network structure is that it allows for parallel computation.

In this paper, the self-attention mechanism is used to calculate the correlation between time steps in the taxi flow input matrix **X** by three vectors: **Query**(**Q**), **Key**(**K**) and **Value**(**V**). The input matrix **X** of the Transformer layer, which is also the output matrix of the LSTM layer, is a (10 × d) matrix. The calculation formulas are as follows:

$$\mathbf{Q} = \mathbf{X}\boldsymbol{W}_{\mathrm{Q}}, \mathbf{Q} \in \mathbb{R}^{10\times \mathrm{d}} \quad (8)$$

$$\mathbf{K} = \mathbf{X}\boldsymbol{W}_{K}, \mathbf{K} \in \mathbb{R}^{10\times \mathrm{d}} \quad (9)$$

$$\mathbf{V} = \mathbf{X}\boldsymbol{W}_{V}, \mathbf{V} \in \mathbb{R}^{10\times \mathrm{d}} \quad (10)$$

$$S = \mathrm{Softmax}\left(\frac{\mathbf{Q}^{\mathrm{T}}\mathbf{K}}{\sqrt{d_K}}\right) \quad (11)$$

$$\mathbf{Z} = \mathbf{V}S^{\mathrm{T}} \quad (12)$$

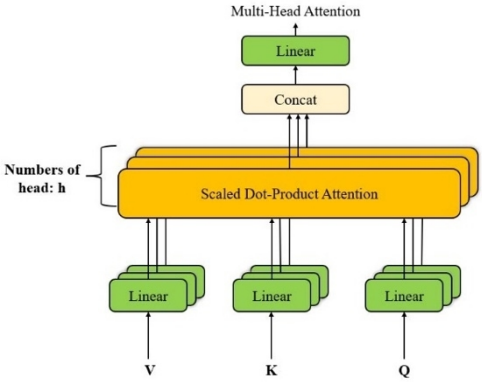


Figure 3: Multi-head self-attention mechanism.

Where $\mathbf{W_Q}$, $\mathbf{W_K}$ and $\mathbf{W_V}$ are weight matrices; d_K is the vector dimension of matrix **K**; $\mathbf{S} \in \mathbb{R}^{10\times d}$ is the similarity matrix between **Q** and **K**; **Z** is the output of the self-attention mechanism.

In addition, we introduce the multi-head self-attention mechanism to use each set of attention to map the input to a different sub-representation space, which allows the model to exploit the periodicity features of the taxi flow more adequately. The structure of the multi-head self-attention mechanism is shown in the Fig. 3.

In the model, **h** self-attention heads are set to be computed in parallel. The output of each self-attention head is stitched horizontally in the computation process to obtain a (10×d×h) matrix, which is multiplied with the mapping matrix to obtain. The mapping matrix can fuse all the traffic timing information, and the output matrix has the same dimension as the input matrix, which can contain the output information of each self-attention head. The calculation formula is as follows.

$$\mathbf{Z}_{\mathrm{M}} = \mathrm{Contact}(\mathbf{Z}_1, \mathbf{Z}_2, \ldots, \mathbf{Z}_i, \ldots, \mathbf{Z}_h)\boldsymbol{W}_O \tag{13}$$

Where Contact(·) is the matrix splicing function; $\mathbf{Z}_i$ denotes the output of the i th self-attended header; $\boldsymbol{W}_O \in \mathbb{R}^{[d\times h]\times d}$ is the mapping matrix.

DATA DESCRIPTION AND PRE-PROCESSING

The dataset used in this paper is from the North Taxi Storage Yard at Hongqiao High-Speed Railway Station, Hongqiao Hub, Shanghai. The dataset describes the daily taxi departures of the North Storage Yard at a granularity of 3 minutes, and the data from 8:00-24:00 per day from January 1, 2021, to June 31, 2021, at the North Storage Yard are selected for this experiment. To meet the requirement of real-time prediction, the experiment was divided into 96-time steps per day with a granularity of 15 minutes. In terms of pre-processing, the data need to be normalized before being input to the model, and this paper uses the Max-min method to deflate the data to the [0, 1] interval, and the formula for the normalization of Max-min is shown

in (14).

$$X_{norm} = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \tag{14}$$

Where X_{norm} denotes the normalized value, $X_{\max}$ denotes the maximum value of the data, $X_{\min}$ denotes the minimum value of the data, and X denotes the original value. Finally, the preprocessed data are divided into a training set Train_data, a testing set Test_data, which are used for model training and validation respectively.

MODEL EVALUATION METRICS

The time series forecasting model is a regression model, so the two evaluation indicators, MAE (Mean Absolute Error) and RMSE(Root Mean Square Error), are chosen to evaluate the forecasting performance of each model. MAE can be used to assess the similarity between the predicted and true values, and the smaller the mean absolute value error, the better the model fits the data; RMSE indicates the sample standard deviation of the deviation between the predicted and true values. The two evaluation indicators are calculated as follows.

$$MAE = \frac{1}{m}\sum_{i=1}^{m}(|y_i - y_{pre}|) \tag{15}$$

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_{pre} - y_i)^2} \tag{16}$$

Where y_i is the true value of the i th test sample, y_{pre} is the predicted value of the i th test sample, and m is denoted as the number of test samples.

MODEL ENVIRONMENT

The experimental model development environment is GPU: NVIDIA GeForce GTX 1050 Ti, memory RAM: 8GB, system: Windows 10, language: Python 3.7, IDE: Anoconda3 spyder. the model is built based on the deep learning framework Pytorch, while setting the model's batch_size The model is built based on the deep learning framework Pytorch, and the model's batch_size is set to 32, the model loss function is MSE, the learning rate is 0.005, the optimizer is selected as Adam, and the activation functions are all ReLU functions.

MODEL PARAMETER TUNING

The model parameters tuning session was conducted using the taxi north storage yard data for the input time step **d**, the number of LSTM layers, $\boldsymbol{L}_1$ the number of hidden layers **n** and the number of self-attention heads **h**. The range of each parameter was selected within a certain range of values: **d**=(2,4,6,8,10); **n**=(32,64,128); $\boldsymbol{L}_1$ =(1,2,3); **h**=(2,5,10). The experiments

of tuning use evaluation metrics MAPE(and RMSE to determine the effect of different parameter values on the prediction accuracy of the model. The tuning session uses the control single variable method, i.e., only one parameter is adjusted at a time and the other parameters remain unchanged, and after finding the optimal value, the value is fixed and the tuning of the next parameter is carried out.

First, the input time step **d** is tuned. The results are obtained for each value of **d**=(2,4,6,8,10) by traversing the Table 1 and Fig. 4 from Table 1, the prediction error of the model is minimized when the input step **d** = 6.In Fig. 5, as **d** increases from 2 until 6, the error of the model keeps decreasing to 8 when the error is minimum, and as **d** exceeds 6 the error of the model gradually increases again. Therefore, the model performs best when the input step **d** = 8, and this parameter will be used in the next step for tuning the other parameters.

Next, the number of LSTM layers L_1, the number of hidden neurons **n**, and the number of self-attention heads **h** are tuned as above. The experimental results are shown in Fig. 5 and Table 2.

From Table 2, we can see that for LSTM hidden layers **n**, the model has the best prediction effect when **n**=32 at the beginning, and the model accuracy keeps decreasing with the increase of **n**. Although theoretically increasing the number of hidden layers will improve the model effect, too many will lead to model overfitting and affect the model accuracy. For the number of LSTM layers L_1, the model accuracy increases when L_1 increasing from 1 to 2, but decreases when L_1 increasing from 2 to 3, indicating that the number of LSTM layers L_1 is sensitive to the model. Increasing or Decreasing one layer will significantly affect the model accuracy, so when L_1=2, the model

Table 1. Results of parameter d tuning.

Parameter	MAE	RMSE
d		
2	25.50	34.87
4	25.16	34.16
6	**25.02**	**33.54**
8	25.42	34.01
10	27.83	37.18

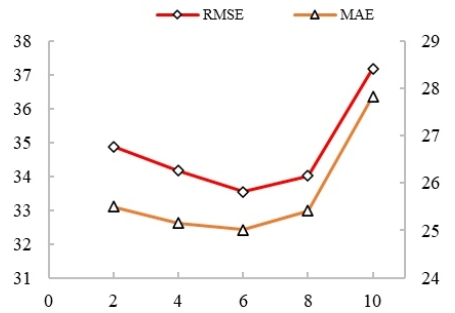
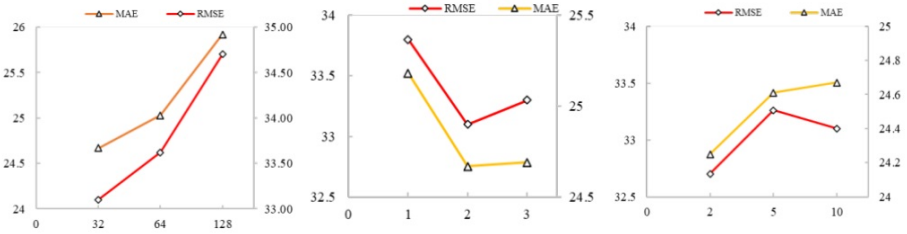


Figure 4: Results of parameter d tuning.

Table 2. Results of parameter **n**, L_1, **h** tuning.

Parameter			MAE	RMSE
n	L_1	h		
32	**2**	**10**	**24.67**	**33.10**
64	2	10	25.03	33.62
128	2	10	25.92	34.70
32	1	10	25.18	33.80
32	3	10	24.69	33.30
32	2	5	24.61	33.26
32	**2**	**2**	**24.25**	**32.70**



a) Parameter **n** b) Parameter L_1 c) Parameter **h**

Figure 5: Process of parameter **n**, L_1, **h** tuning.

accuracy is the best. For the number of Transformer self-attention heads **h**, the model accuracy is the highest when **h**=2 at the beginning, and the model accuracy keeps decreasing as increasing to 5 and 10, indicating that the number of self-attention heads has a great influence on the model prediction. Moreover, the number of self-attention heads needs to be divisible by output_size = 10 when setting, so the setting value can only be 2, 5 and 10, which has some limitations and affects the model accuracy. In conclusion, the optimal combination of parameters for the LSTM-Transformer model, i.e., **d**=6, **n**=32, L_1=2, **h**=2, was finally determined, at which the evaluation metrics MAE = 24.25 and RMSE = 32.70 is at most 6.4% and 5.8% lower than the MAE and RMSE values of other parameter combinations. The Fig. 6 shows the prediction effect of the model for the first 250 time steps of the dataset. It can be seen that the model captures the temporal characteristics of the dataset well and has a good prediction effect.

MODEL PERFORMANCE COMPARISON

To verify that the proposed Transformer-LSTM model has excellent performance, the dataset of the taxi north storage yard is selected for the experiments, and the obtained results are compared with GRU, CNN, and LSTM models. The experimental results are shown in Table 3.

From Table 3, it can be seen that the LSTM-Transformer model has the lowest MAE and the second lowest RMSE for the prediction of the dataset,

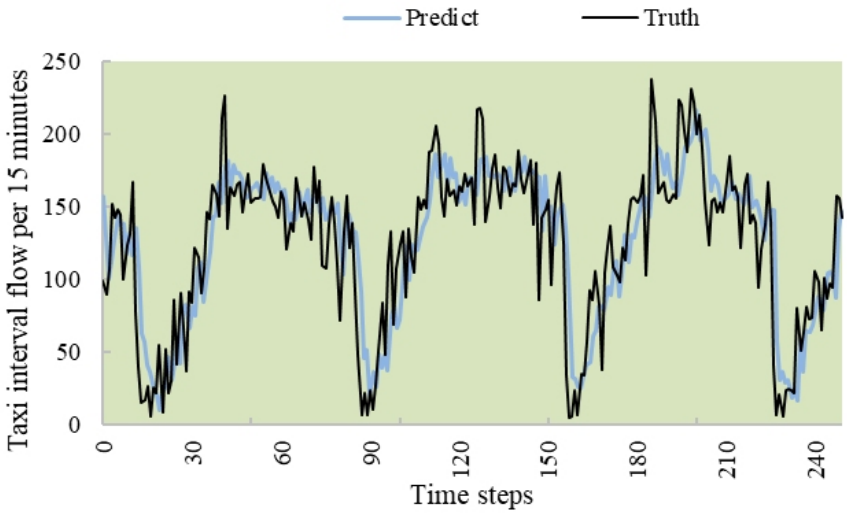


Figure 6: Part of predicted and actual results.

Table 3. Model comparison results.

Model name	MAE	RMSE
LSTM-Transformer	**24.25**	32.70
CNN	32.61	39.23
GRU	24.78	**31.73**
LSTM	28.56	35.79

and the RMSE of the LSTM-Transformer model is 16.6% and 8.6% lower than the other two baseline models, CNN and LSTM, respectively, while the MAE is also 25.6% and 15.1% lower than them, respectively,which can indicate that the error between the prediction results of our proposed model and the true value is smaller, and the prediction accuracy is significantly improved compared with other models. On the other hand, the experimental results show that GRU has the lowest RMSE and the second lowest MAE in terms of prediction accuracy. This shows that the LSTM-Transformer has very good accuracy in prediction on the dataset, and the LSTM-Transformer accuracy is similar to the GRU model. It is worth mentioning that GRU, as an excellent recurrent neural network for processing time series, has good accuracy in predicting regression problems.

CONCLUSION

To improve the operational management efficiency of transportation hubs, a combined model of taxi short-time prediction based on the LSTM neural network and Transformer model is proposed. Model comparison experiments were conducted using three baseline models, CNN, LSTM, and GRU, and the experimental results showed that the LSTM-Transformer model has

good accuracy compared with other models, while the accuracy of the LSTM-Transformer model is similar to the GRU model. The demand for taxis is affected by other modes of transportation and external uncertainties out of the hub, and these problems are not considered in this paper. In the subsequent study, we will consider the influence of more factors and the traffic patterns of the hub to conduct a more in-depth study of taxi demand forecasting problem.

REFERENCES

Chen, E., Ye, Z., Wang, C., Xu, M., (2019). Subway Passenger Flow Prediction for Special Events Using Smart Card Data. IEEE Transactions on Intelligent Transportation Systems, 21, 1109–1120.

Han, Y., Wang, C., Ren, Y., Wang, S., Zheng, H., Chen, G., (2019) Short-Term Prediction of Bus Passenger Flow Based on a Hybrid Optimized LSTM Network. International Journal of Geo-Information, 8(9), 366.

Hochreiter S., Schmidhuber J., (1997). Long short-term memory. Neural computation, 9(8), 1735–1780.

Jiao, P., Li, R., Sun, T., Hou, Z., Lbrahim, A., (2016). Three revised Kalman filtering models for short-term rail transit passenger flow prediction. Mathematical Problems in Engineering, 2016(3), 1–10.

Lin, L., Liu, X., Liu, X., Zhang, T., Cao, Y., (2022). A prediction model to forecast passenger flow based on flight arrangement in airport terminals. Energy and Built Environment, S2666123322000423.

Liu, T., Wu, W., Zhu, Y., Tong, W., (2020). Predicting taxi demands via an attention-based convolutional recurrent neural network. Knowledge-Based Systems, 206, 106294.

Rodrigues, F., Markou, I., Pereira, F. C., (2019) Combining time-series and textual data for taxi demand prediction in event areas: A deep learning approach. Information Fusion, 49, 120–129.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., Polosukhin, I., (2017). Attention Is All You Need. arXiv.

Wang, S., Zhao, J., Shao, C., Dong, C. D., Yin, C., (2020). Truck Traffic Flow Prediction Based on LSTM and GRU Methods With Sampled GPS Data. IEEE Access, 8, 208158-208169.

Yang, D., Chen, K., Yang, M., Zhao, X., (2019). Urban rail transit passenger flow forecast based on LSTM with enhanced long-term features. IET Intelligent Transport Systems, 13(10), 1475–1482.

Zhong, M.and Liu, Y., (2020). Research on the Evaluation Index System of the Taxi Operation and Service in Airport Hubs, MATEC Web of Conferences, 325, 04005.