

Article

Short-Term Travel Demand Prediction of Online Ride-Hailing Based on Multi-Factor GRU Model

Qianru Qi ^{1,2,3}, Rongjun Cheng ^{1,2,3,*} and Hongxia Ge ^{1,2,3}

¹ Faculty of Maritime and Transportation, Ningbo University, Ningbo 315211, China; 2011084068@nbu.edu.cn (Q.Q.); gehongxia@nbu.edu.cn (H.G.)

² Jiangsu Province Collaborative Innovation Center for Modern Urban Traffic Technologies, Nanjing 210096, China

³ National Traffic Management Engineering and Technology Research Centre, Ningbo University Sub-Centre, Ningbo 315211, China

* Correspondence: chengrongjun@nbu.edu.cn

Abstract: In recent years, online ride-hailing has become an indispensable part of residents' travel mode. Therefore, the prediction of online ride-hailing travel demand has become extremely important. In the era of big data, the application of big data in the field of transportation is becoming more extensive. Based on the open data of ride-hailing trips in Haikou City, Hainan Province, provided by the Didi platform and combined with the rainfall data of Haikou City, this paper proposes a gate recurrent unit (GRU) model considering rainfall factors and rest days factors for short-term trip demand prediction. The K-fold cross-validation method is adopted to adjust the parameters of the model to the optimal ones through the training set. The improved GRU model is compared with the original GRU model and other classic models, and the model is evaluated by root mean square error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE), and *R*² score indexes. Finally, it is proved that the GRU model proposed in this paper greatly improves the prediction accuracy of short-term online ride-hailing travel demand.

Keywords: GRU; rainfall intensity; multiple factors; online ride-hailing travel demand prediction; big data



Citation: Qi, Q.; Cheng, R.; Ge, H. Short-Term Travel Demand Prediction of Online Ride-Hailing Based on Multi-Factor GRU Model. *Sustainability* **2022**, *14*, 4083. <https://doi.org/10.3390/su14074083>

Academic Editors: Dianhai Wang, Zhenyu Mei, Lihui Zhang and Xiaofei Ye

Received: 24 February 2022

Accepted: 28 March 2022

Published: 30 March 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In the era of big data, the use of big data in the transportation industry is becoming more extensive. Online ride-hailing has become popular across the country and is an important means of transportation for people because online ride-hailing is more convenient and flexible than bus and metro for passengers. The travel demand prediction of online ride-hailing has great significance for the development of online ride-hailing [1–4]. Deep learning is a new research direction in the field of machine learning, which is now widely applied in the field of transportation. Through multi-layer processing, the initial “low level” features are gradually transformed into “high level” features, and the complex learning tasks can be completed with simple models. In 1994, Bengio et al. [5,6] looked into this problem in depth, and found some fairly fundamental factors that make training recurrent neural networks (RNNs) very difficult. Then, in 1997, Schmidhuber et al. [7] proposed a long short-term memory (LSTM) neural network, a special RNN, which solves the problem of long-term dependence of RNNs and can learn long-term dependence information. In recent years, in the field of transportation, many scholars have used the LSTM model to predict traffic flow. Luo et al. [8] proposed a spatio-temporal traffic flow prediction method combining K-nearest neighbor (KNN) and LSTM. KNN is used to select the adjacent stations that have the closest relationship with the test stations to capture the spatial characteristics of the traffic flow. LSTM is used to mine the temporal variability of the traffic flow, and a two-layer LSTM network is used to predict the traffic flow of the

selected stations. Yang et al. [9] proposed a method to improve LSTM by capturing these high-impact traffic flow values using an attention mechanism. At the same time, some data beyond the normal range were smoothed to obtain better prediction results. The results showed that the prediction model has certain competitiveness in short-term traffic flow prediction. The gate recurrent unit (GRU) can also be seen as a variant of LSTM, as their underlying concepts are similar and produce equally impressive results in some cases. Lu et al. [10] used the data of Didi and Conv-LSTM to forecast the travel demand of online ride-hailing. Reasonable prediction results can provide data support for vehicle scheduling and distribution, solve problems such as energy waste and traffic congestion caused by asymmetry between supply and demand, and maximize the benefits of passengers, drivers, and ride-hailing platforms. In 2014, Cho et al. [11] proposed the GRU model, which consists of a two-layer RNN. Wang et al. [12] used LSTM and GRU to forecast vehicle traffic flow with GPS sampling data in the road network. The results showed that the LSTM and GRU methods have better prediction performance than the existing autoregressive integrated moving average (ARIMA) methods. Dai et al. [13] proposed a short-term traffic flow prediction model combining spatiotemporal analysis and GRU. The prediction results of the proposed model were compared with the actual traffic flow data to verify the effectiveness of the model. Li et al. [14] combined the transfer learning method to solve the problem of insufficient online traffic data, adopted the particle filter online training algorithm to reduce the training time complexity, and used the GRU model to achieve accurate prediction of satellite network traffic.

Many factors, such as weather conditions, morning and evening rush hours, and holidays, can affect travel. Therefore, combining various factors to forecast the traffic flow can make the prediction result more accurate [15–17]. Zhang et al. [18] used GRU model to predict traffic flow and combined weather factors to make the prediction results more accurate. Li et al. [19] used different deep-learning models to forecast pedestrian travel in hazy weather. Liu et al. [20] proposed an hourly passenger-flow prediction model based on deep learning by using stacked autoencoders (SAE) and a deep neural network (DNN). Travel characteristics, defined as time characteristics, scene characteristics, and passenger flow characteristics are inputs in this model. Additionally, experimental results showed that this method can provide a more accurate and general passenger-flow prediction model for bus rapid transit (BRT) stations with different passenger-flow distributions. Table 1 summarizes the current literature on online ride-hailing travel demand forecasting.

Table 1. Literature on online ride-hailing travel demand forecasting.

Author, Year	Model	Consider Influence Factor	Consider Multi-Factor
Nejadettehad et al. [21] (2020)	RNN	No	No
Jin et al. [22] (2020)	Stacking ensemble learning	Yes	No
Tian et al. [23] (2020)	CNN-LSTM	Yes	No
Li et al. [24] (2020)	ST-transformer	Yes	No
This paper	GRU	Yes	Yes

There are not much research in the field of online ride-hailing travel demand prediction, and there is no deep-learning model based on multi factors in the prediction of online ride-hailing travel demand. Based on the research of the above scholars, in order to fill the gap in travel demand prediction of online ride-hailing and multi-factor prediction, this paper proposes a GRU model with multi-factor consideration to predict travel demand of online ride-hailing. Based on analysis of the data characteristics, this paper proposes a GRU model considering multiple factors. The weight parameters are adjusted to the optimal by K-fold cross validation method, and then the model proposed in this paper is evaluated by various evaluation functions and compared with other models. The purpose of this paper is to optimize the supply relationship of online car-hailing and make full use of online car-hailing resources according to the prediction of online car-hailing travel demand. This

can reduce road traffic pressure and traffic pollution, which is of great significance to the sustainable development of the city.

2. Data Analysis

2.1. Data Sources and Processing

The data selected in this paper are from the online ride-hailing trip data in Haikou of Didi Chuxing GAIA Initiative (<https://gaia.didichuxing.com>, accessed on 26 May 2021). “Didi Chuxing” has changed the traditional taxi mode and established a modern travel mode for users in the era of the big mobile internet. Compared with traditional telephone call and curb call, the birth of the drip taxi has changed the traditional taxi-market pattern, overturning the concept of curb-block. Using the characteristics of mobile internet, a combination of online and offline will maximize the passenger taxi experience, change the traditional ways of taxi drivers and other passengers, save the cost of communication between drivers and passengers, reduce the empty rate, and maximize savings of resources and time on both sides of the company. As shown in Figure 1, the area enclosed by red line is the area of Haikou City. In this paper, we have used the Haikou online ride-hailing trip data of 1 May 2017 to 1 June 2017 and the hourly rainfall-intensity data of Haikou City in May, obtained. through Python web crawler. Next, the online ride-hailing trip characteristics of that month will be analyzed.

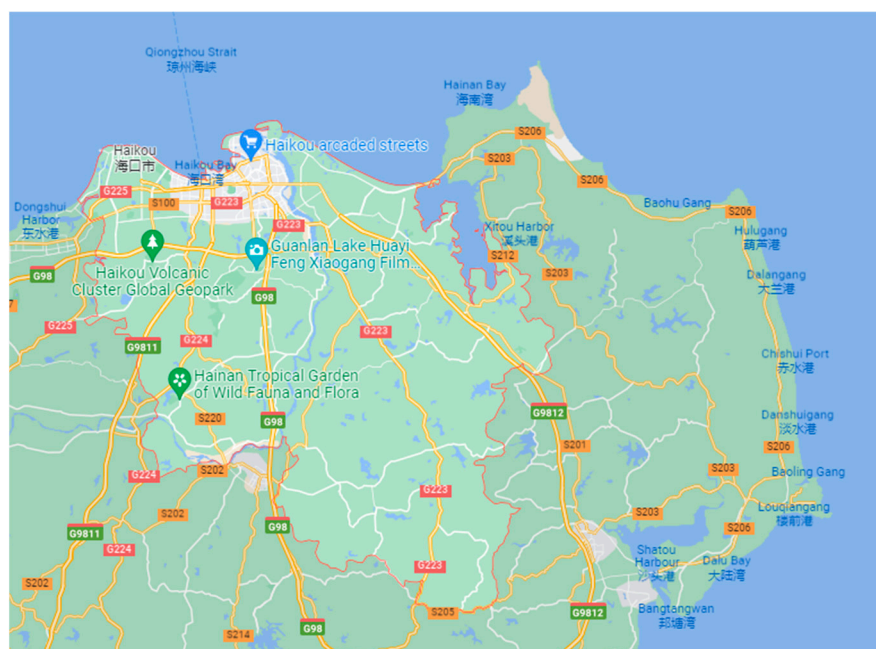


Figure 1. Scope map of Haikou City.

Firstly, the data was cleaned. Data cleaning refers to the final procedure of finding and correcting identifiable errors in data files, including checking data consistency and dealing with invalid and missing values. Data cleaning after entry is usually done by computers rather than by humans. For this paper, there were more than two million pieces of online car-hailing data in the month. As a medium-sized data storage platform, SQL Server has the advantages of efficient data processing and flexible background development, which can analyze the data better. With the SQL database, invalid travel times, longitude, latitude and other missing values are removed, which will ensure the accuracy of data.

Then, the hourly travel volume of online ride-hailing is calculated by SQL language, and the hourly travel volume of each day is summed to give the total travel volume of each day. The number of online ride-hailing trips per day in a month is shown in Figure 2.

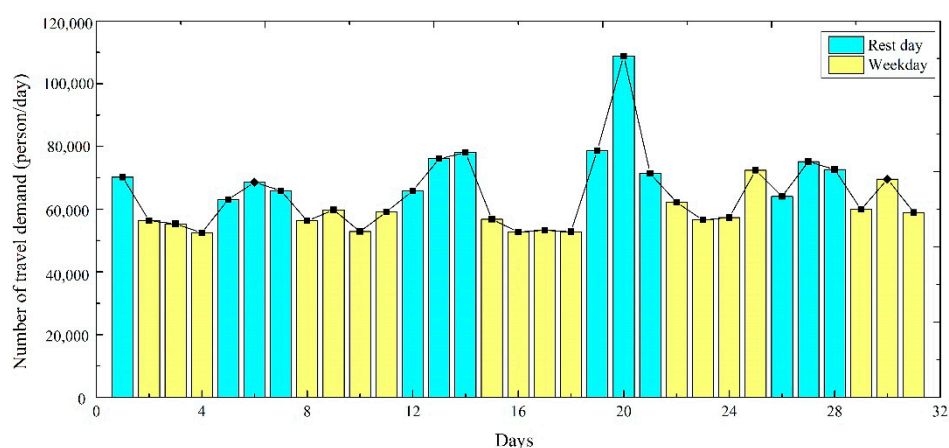


Figure 2. Daily number of online ride-hailing trips.

The chart above shows rest days in blue, while the yellow shows weekdays. Since the travel characteristics of Friday after 2 p.m. are similar to weekends, Friday is divided into rest days to set the weight. Setting different weights on rest days and working days to separate them can help the model gain weight information during learning and make the prediction more accurate. It can be clearly seen that the travel volume of online ride-hailing on rest days was significantly greater than that on weekdays, indicating that the travel demand of online ride-hailing on rest days is stronger than that on weekdays.

2.2. Rainfall Data Analysis

China's meteorological departments classify rainfall according to its intensity [25]. Rainfall intensity refers to the amount of rainfall per unit period of time, which is generally expressed by the depth of rainfall per unit time. According to the difference in unit time, there are several different classification methods of rainfall levels; meteorological departments generally set the unit time as 1 h, 12 h, and 24 h. The classification standards of rainfall grades in China are shown in Table 2:

Table 2. Classification standards of rainfall grades in China.

Rainfall Level	1 h Rainfall (mm)	12 h Rainfall (mm)	24 h Rainfall (mm)
Light rain	≤2.5	≤4.9	≤9.9
Moderate rain	2.6–8	5–14.9	10–24.9
Heavy rain	8.1–16	15–29.9	25–49.9
Torrential rain	≥16.1	≥30	≥50

Raw meteorological data include important parameters such as hourly atmospheric pressure, temperature, mean relative humidity, mean wind speed, and rainfall. The influence of factors other than rainfall on travel demand is small, so is not considered. This paper focuses on the influence of rainfall on online ride-hailing travel demand characteristics, so only hourly rainfall data and the date and time of the corresponding data collection were extracted. The longitude and latitude of Haikou meteorological station selected in this paper is 110.25° and 20°. In order to facilitate the research, it was assumed that rainfall was uniformly distributed during the data collection period.

Additionally, combined with rainfall intensity and online ride-hailing travel data, the comparison of hourly travel demand was made between 19 May 2017 (Friday with rain), 26 May 2017, and 12 May 2017 (both Friday with no rain). As shown in the figure below, the black column indicates the intensity of rainfall, while the black broken line indicates the number of online ride-hailing trips per hour on 19 May.

It can be seen in following Figure 3 that, on this day, the rainfall intensity at 17:00 and 18:00 in the evening was relatively high, reaching the level of moderate rain. Compared with the other two days, the travel demand for online ride-hailing at this time also significantly

increased. According to the travel characteristics of online ride-hailing, this paper proposes an improved model based on the GRU model, which takes into account the factors of rainfall intensity and rest days.

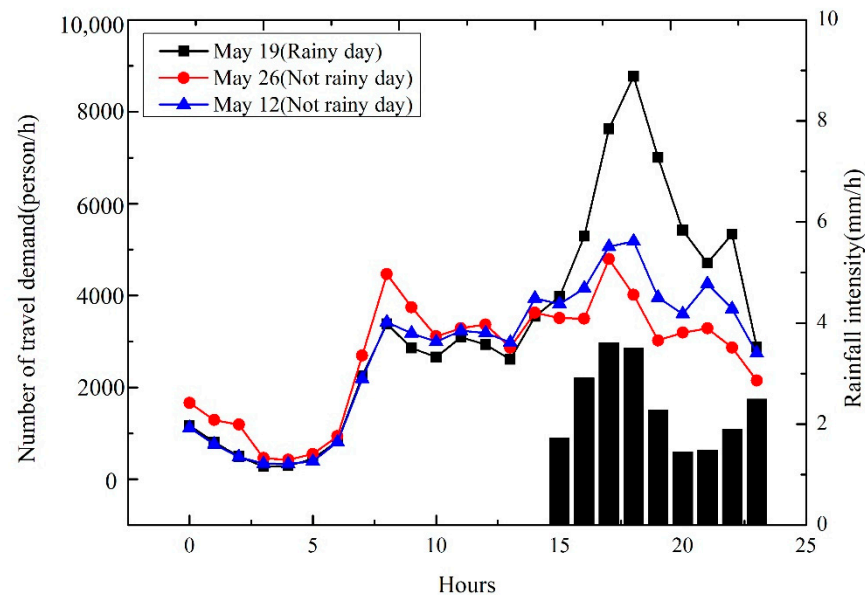


Figure 3. Comparison of online ride-hailing travel demand combined with rainfall intensity.

3. Model Analysis

3.1. Gate Recurrent Unit

The principle behind the GRU is very similar to LSTM, that is, it uses a gating mechanism to control input, memory, and other information and make predictions at the current time-step [11]. The GRU has two gates, a reset gate and an update gate. Intuitively, the reset gate determines how new input information is combined with previous memories, and the update gate defines the number of previous memories to be saved to the current time-step. Basically, these two gated vectors determine what information can ultimately be used as the output of the gated loop unit. These two gating mechanisms are unique in that they are able to preserve information in long-term sequences without being cleared over time or removed because it is not relevant to the prediction.

As shown in Figure 4 above. The update gate helps the model determine how much information from the past should be transferred to the future, or how much information from the previous time-step and the current time-step should be transferred. This is very powerful because the model can decide to copy all the information from the past to reduce the risk of the gradient disappearing. In the time-step t , firstly, we need to calculate the update gate z_t using the following formula:

$$z_t = \sigma(x_t W^z + h_{t-1} U^z) \quad (1)$$

where x_t is the input vector of the time t step, the t component of the input sequence X , which will undergo a linear transformation (multiplied by the weight matrix W^z). h_{t-1} holds information about the previous time-step $t - 1$, and it also goes through a linear transformation. The update adds these two pieces of information into the *sigmoid* activation function, thus compressing the activation result to between 0 and 1. The *sigmoid* function is similar to the *tanh* function and is also non-linear, except that the *sigmoid* function compels the value into a range of 0 to 1, which helps update or forget information. The *sigmoid* function formula is as follows:

$$\text{Sigmoid}(x) = \frac{1}{1 + e^{-x}} \quad (2)$$

Essentially, the reset gate determines how much past information needs to be forgotten, which we can calculate using the following expression:

$$r_t = \sigma(x_t W^r + h_{t-1} U^r) \tag{3}$$

This expression is the same as the update gate expression, except that the parameters and uses of the linear transformation are different. As described in the previous update gate, h_{t-1} and x_t first undergo a linear transformation, and then add the *sigmoid* activation function to output the activation value.

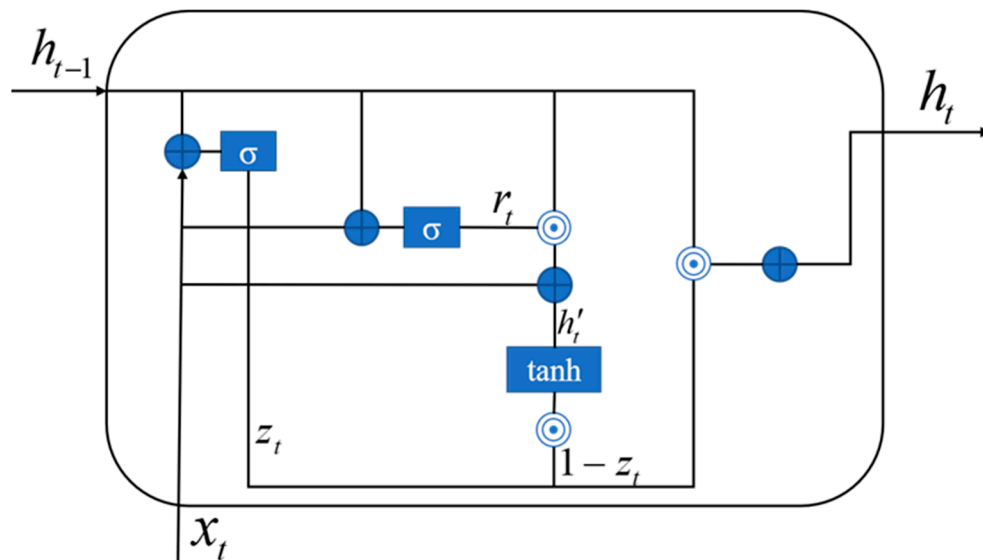


Figure 4. GRU model structure.

We will now discuss, in detail, how these gates affect the final output. In the use of the reset gate, the new memory content will use the reset gate to store information related to the past. Its calculation expression is:

$$h'_t = \tanh(x_t W^h + r_t \odot U^h h_{t-1}) \tag{4}$$

The input x_t and the previous time-step information h_{t-1} first undergo a linear transformation, namely, the right multiplication of the matrices W^h and U^h , respectively. The Hadamard product of reset gate r_t and $U^h h_{t-1}$ is calculated, that is, the product of corresponding elements of r_t and $U^h h_{t-1}$. Because the reset gate calculated earlier is a vector of 0 to 1, it measures the size of the gate opening. For example, a gate value of 0 for an element means that information about that element has been completely forgotten. The Hadamard product determines what previous information is to be retained and forgotten. The results of these two parts are then added and put into the hyperbolic tangent activation function.

In the last step, the network needs to compute the h_t vector, which will hold the information of the current cell and pass it to the next cell. In this process, we need to use the update gate, which determines what information needs to be collected in the current memory content h'_t and the previous time-step h_{t-1} . This process can be expressed as:

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot h'_t \tag{5}$$

where z_t is the activation result of the update gate, which also controls the inflow of information in the form of gate control. The Hadamard product of z_t and h_{t-1} represents the information retained by the previous time-step to the final memory. This information, together with the information retained by the current memory to the final memory, is equal to the content output by the final gated loop unit.

In Equations (1)–(5), W^z , W^r , W^h , U^z , U^r and U^h are the weight of each calculation layer and h_{t-1} is the output of time $t - 1$. In addition, h_t and x_t represent the output and the input of time t .

3.2. Improved Model

GRU model has good accuracy in the case of less training times or small data sample size, so the GRU model was adopted based on the sample size in this paper. As shown in Figure 5 below, the technical roadmap for the improved model proposed in this paper is as follows:

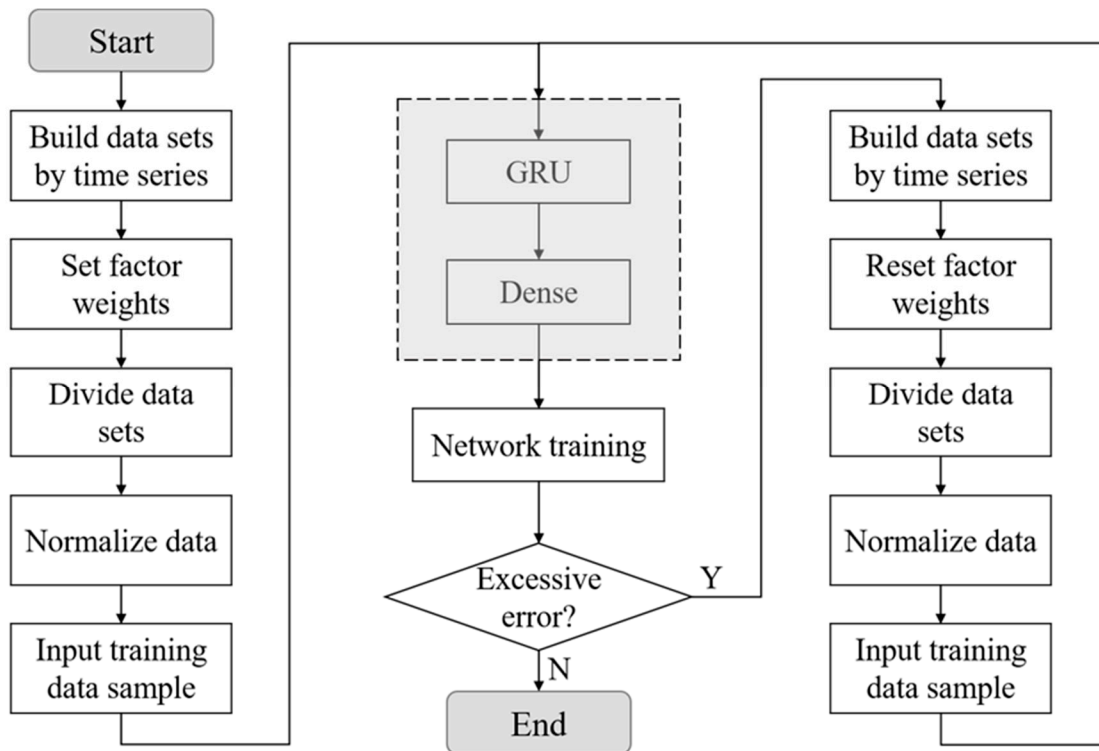


Figure 5. Model training process.

Step 1: The data sets are built by time series. The GRU model is a time-series model, which requires the data set to have a stable time-series, otherwise the time-series model cannot be established. The data spans from the 1 May 2017 solstice to 1 June 2017, in which, travel data of a day is calculated according to the hourly travel demand.

Step 2: The factor weights are set. In this paper, two influence factors of online ride-hailing are selected. One is the rainfall intensity, the other is the off-day factor. The weight of rainfall intensity factor is set according to millimeters of rainfall per hour. The weight of seven days a week is set according to the data analysis above. The weight of Monday to Friday at 2:00 p.m. is 0.2, and the weight of Friday at 2:00 p.m. to Sunday at 12:00 a.m. is 0.4, among which the weight of holidays is 0.6. Input the two factors and hourly travel demand data into the GRU model at the same time, and the formula is as follows:

$$x_t = (D_h \quad W_r \quad W_d) \quad (6)$$

where x_t is the input, D_h , W_r , and W_d , respectively, represent the hourly travel demand for online ride-hailing, hourly rainfall, and the weight of each hour of the day. The input dimension of this paper is (744, 2, 4), which expresses the number of rows representing the input, which is 744, the number of columns, which is 4, and the number of GRU cells, which is 2.

Step 3: The data sets are divided. The training set is the learning sample data set, which is mainly used to train the model. The verification set is to adjust the parameters

of the classifier on the learned model, such as selecting the number of hidden units in the neural network. Validation sets are also used to determine the network structure or parameters that control the complexity of the model. The test set is mainly to test the resolution (recognition rate, etc.) of the trained model. In this paper, the original data set was divided into training set and test set, and the ratio of training set and test set was 4:1. The data from 1 May 2017 to 1 June 2017 are sorted according to the time series. The first 80% of the data are the training set and the last 20% are the test set. In addition, The data of the validation set is the last 10% of the training set.

Step 4: The data is normalized. The original data maintains its original distribution, and the data values are normalized to a range of 0 to 1. After data normalization, the original features of the data can be well retained, but the size of the data value is reduced, which is of great help to model training and prediction. In this paper, the min-max normalization method is selected to normalize the data. The calculation formula is as follows:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (7)$$

where x' is the normalized data value and x is the real data value.

Step 5: Input the training set into the model for training. According to the training results of the training set, the parameters of the model and the weight of the model input are adjusted constantly. In this paper, the K-fold cross-validation method is used [26–28], and $K = 10$ is taken without repeated sampling. The training set is randomly divided into 10 pieces. Each time, 1 piece is selected as the validation set, and the remaining 9 pieces are selected as the training set. The average value of the 10 test results was calculated as the estimation of the model accuracy and as the performance index of the model under the current K-fold cross-validation. The schematic diagram of K-fold cross-validation is shown in Figure 6 below. If the model error is too large (if $\text{MAPE} > 40\%$ or $R^2 < 0.8$), all parameters are reset and the training set is re-trained. Through continuous training and debugging, the optimal parameters and weights are finally determined.

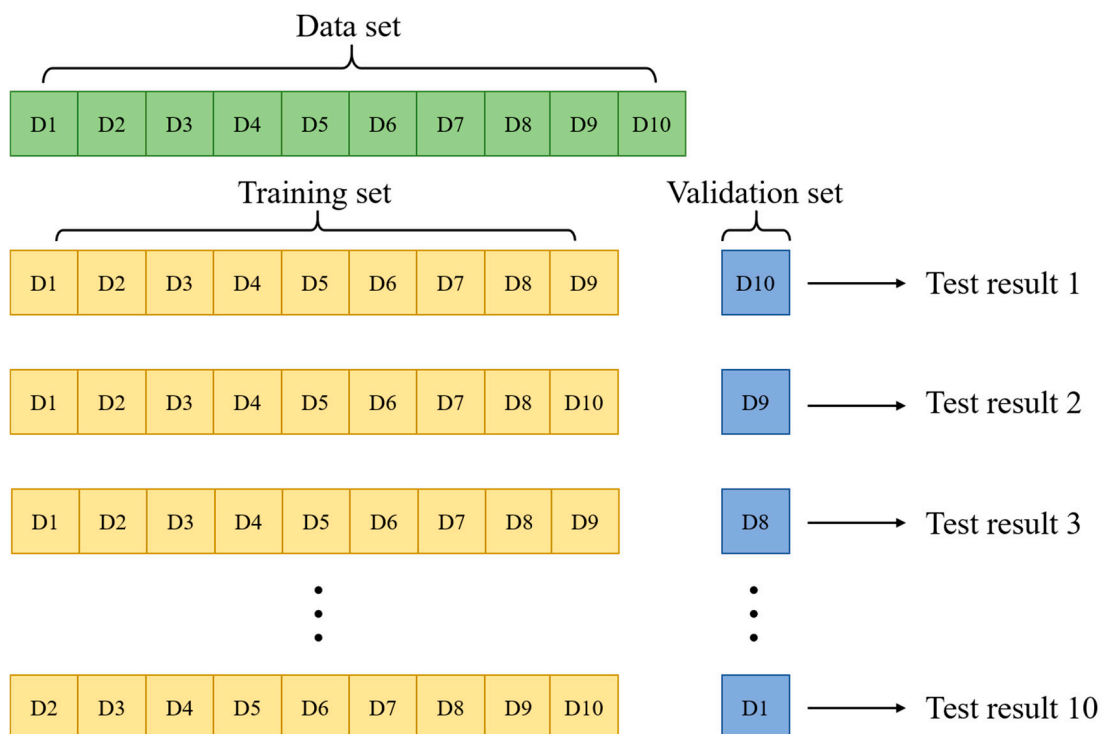


Figure 6. K-fold cross-validation schematic diagram.

Through the training of the training set and the adjustment of the model in the above steps, the final parameters of the model in this paper are determined as follows (Table 3):

Table 3. Detailed description of parameters.

Parameter	Value
Number of hidden layers	2
Number of each hidden layer neurons	24
Training times	50
Activation function of hidden recurrent layers	tanh
Details of weight parameters	Weight from 12:00 a.m. Monday to 2:00 p.m. Friday is 0.4
	Weight from 2:00 p.m. Friday to 12:00 a.m. Sunday is 0.6
	Weight of holidays is 0.8
	Rainfall weights are based on actual rainfall intensity
Backstep	24

“Number of hidden layers” in Table 3 above indicates the number of hidden layers of this model. “Number of each hidden layer neurons” indicates the number of neurons in each hidden layer of the model in this paper. “Training times” indicates the number of training times in a round of model training. “Activation function of hidden recurrent layers” indicates the activation function selected by the model. In the selection of activation function, combined with previous studies, we found that the activation functions of most GRU models use tanh, relu, and *sigmoid* functions. When the input of tanh is large or small, its output is smoother and the gradient is smaller. Then, under the same model conditions, we have compared the effects of tanh and relu activation functions, and the result is that tanh has a better effect. “Backstep” indicates the number of previous data learned during model training.

In this paper, the learning rate of the model is set by the adaptive gradient descent algorithm. The batch gradient descent algorithm is used to process the whole training set at the same time, and all samples are used to update its parameters. Since the number of training sets in this paper is not very large, the code execution speed will not be long when using the batch gradient descent algorithm, and the accuracy will also be improved.

The hourly weight is set according to the characteristics of online car-hailing demand, and the hourly weight is input into the model as a feature quantity for the model to learn.

4. Result Analysis

4.1. Selection of the Evaluation Function

In this paper, root mean square error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE), and decision coefficient R^2 score are selected as the accuracy evaluation indexes to compare these prediction algorithms [29–32].

The root mean square error (RMSE) is the square root of the ratio of the square of the deviation between the predicted value and the true value to the number of observations n . In actual measurements, n is always finite, and the true value can only be replaced by the most reliable (best) value. RMSE ranges from 0 to infinity, and is equal to 0 when the predicted value is in perfect agreement with the real value. The larger the error, the greater the value.

The full name of MAE is mean absolute error, which is the mean of the absolute error. MAE is another commonly used regression loss function. It is the mean of the absolute sum of the difference between the target value and the predicted value. It represents the mean error margin of the predicted value, regardless of the direction of the error, and ranges from 0 to infinity. When the predicted value is exactly consistent with the true value, it is equal

to 0, which is the perfect model. The larger the error, the greater the value. The advantage of MAE is that it is less sensitive to outliers and more stable as a loss function.

The mean absolute percentage error (MAPE) is one of the most popular metrics used to evaluate prediction performance. MAPE ranges from 0 to infinity. When the MAPE value is 0%, it means the model is perfect, while if the MAPE value is greater than 100%, it means the model is poor. The value of the MAPE alone is meaningless because MAPE is a relative value, not an absolute value. The MAPE depends on the size of the numbers in the data. For example, if you have two numbers, one is 100, one is 1, and your predictions are 101 and 2, each with an error of 1, but the MAPE value is bigger and the other is smaller. Similarly, when the actual value is large, the values of RMSE and MAE will also be large. Therefore, it is meaningless to look at the values of RMSE and MAE alone; they need to be compared with other models under the same data.

R^2 refers to goodness of fit, and is the fitting degree of the regression line to the observed value. In deep learning, R^2 is usually called the R^2 score. The R^2 score can be colloquially understood as using the mean as the error base to see if the prediction error is greater than or less than the mean base error. The value of the R^2 score ranges from 0 to 1. An R^2 score of 1 means that the predicted and true values in the sample are exactly the same without any error. In other words, the model we built perfectly fits all the real data, and is the best model with the highest R^2 score. However, the model is not usually this perfect, there is always an error, and when the error is small, the numerator is smaller than the denominator, the model is going to approach 1, which is still a good model, but as the error gets bigger and bigger, the R^2 score is going to get further and further away from the maximum of 1. If the R^2 score value is 0, it means that every predicted value of the sample is equal to the mean value, and the model constructed is exactly the same as the mean value model. If the R^2 score value is less than 0, it indicates that the model constructed is inferior to the benchmark model, and the model should be rebuilt.

In Table 4 and the formulae above, n represents the number of data sample, y_p is the value of forecast data, and y is the actual value.

Table 4. Predictive evaluation index formula.

Metric	Formula
RMSE	$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_p - y)^2}$
MAE	$MAE = \frac{1}{n} \sum_{i=1}^n y_p - y $
MAPE	$MAPE = \frac{100\%}{n} \sum_{i=1}^n \frac{ y - y_p }{ y }$
R^2 score	$R^2 = 1 - \frac{\sum_{i=0}^{n-1} (y_p - y)^2}{\sum_{i=0}^{n-1} (y - \bar{y})^2}$

4.2. The Comparison of Predicted Results

We next analyze the improved GRU model proposed in this paper and the original GRU model for evaluation indexes. The two models are used to predict the same set of data, and the parameter values of the two models are the same as those in Table 3.

As shown in Figure 7 below, the black polyline represents the real data of the test set, while the red and blue polylines represent the test data of the improved GRU model and the original GRU model, respectively. The abscissa of the figure is the data number of the test set, which is the number of data in the prediction set sorted by time series, and the ordinate is the number of travel demand. It is clear that the red line fits better than the blue, which indicates that the improved GRU model predicts better results than the original GRU model.

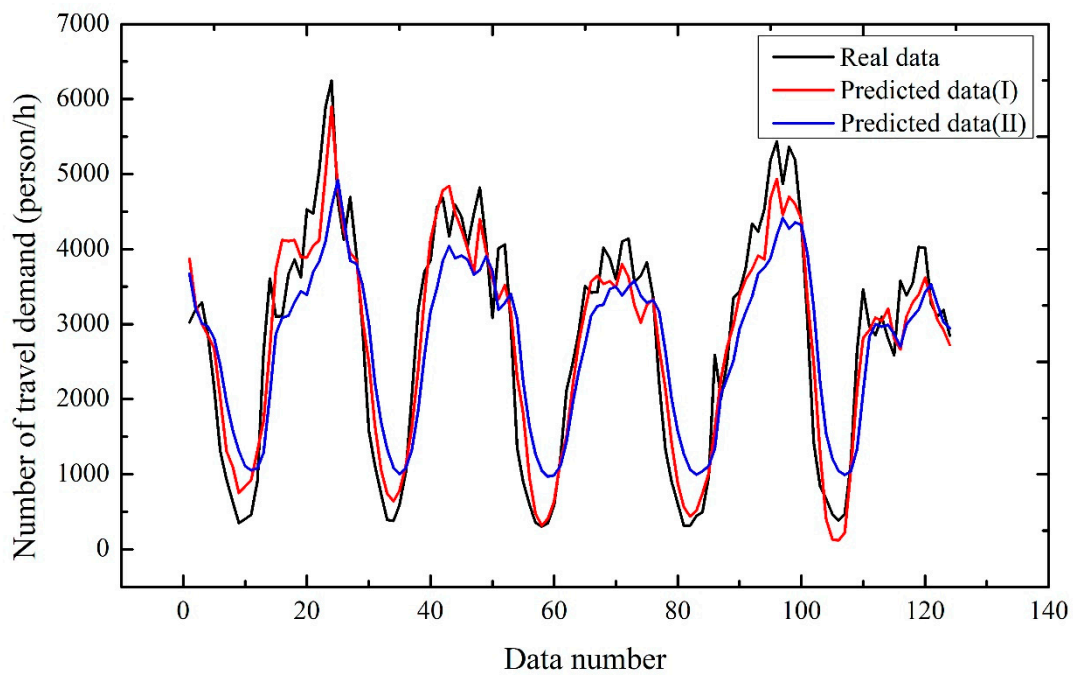


Figure 7. Comparison line chart of the two models.

Figure 8 shows the prediction results of the two models on a rainy day. The figure above selects the data of the rainy day on 30 May in the test set, in which the red poly-line is the improved GRU model, the blue poly-line is the ordinary GRU model, and the black column is the rainfall in that hour. It can be seen that the prediction result of the improved GRU model is better than that of the ordinary GRU model when the rainfall increases sharply.

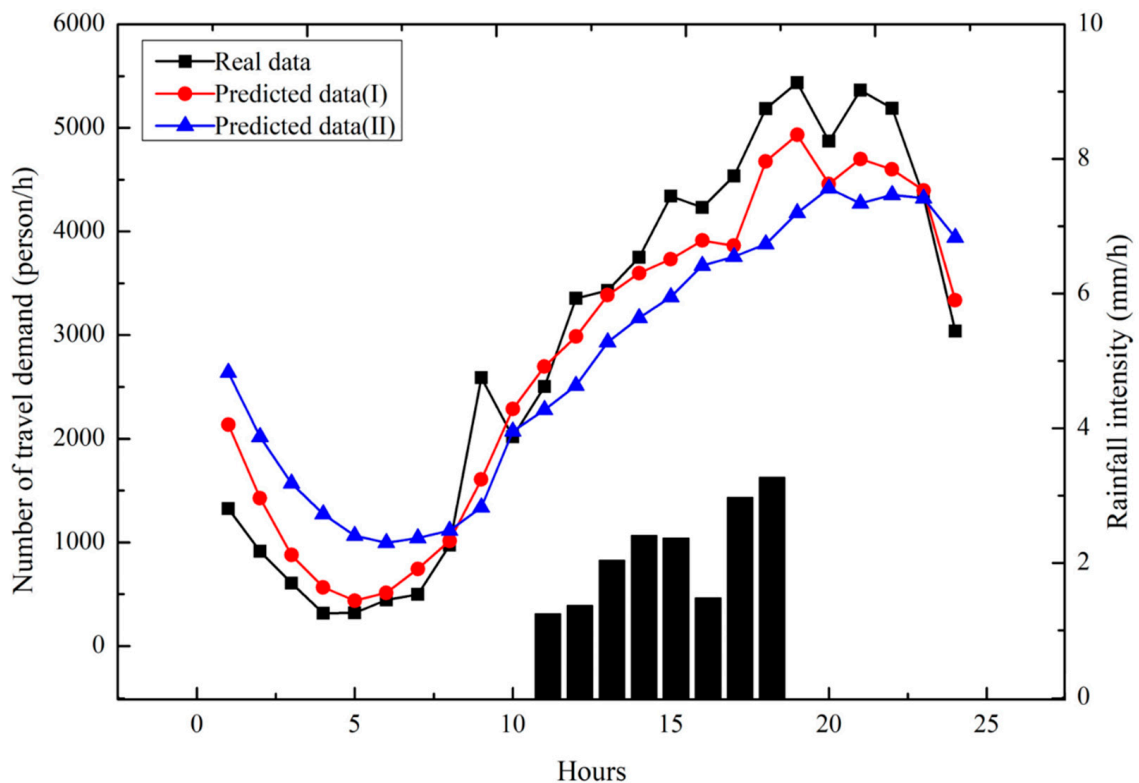


Figure 8. Comparison of predictions for a rainy day.

Figure 9 shows the prediction results of the two models on a rest day. This figure selects the data on Sunday, 28 May. The red polyline is the improved GRU model, and the blue polyline is the ordinary GRU model. It can be seen that the prediction accuracy of the improved GRU model on Sunday is also much higher than that of the ordinary GRU model.

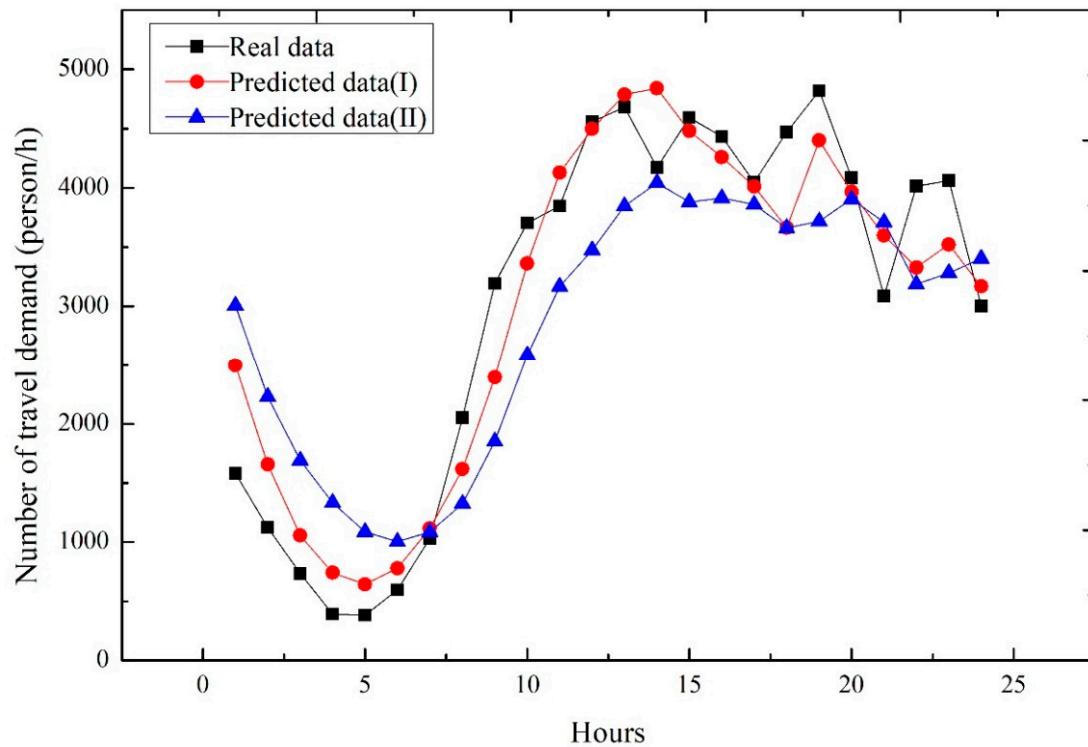


Figure 9. Comparison of predictions for a rest day.

The RMSE, MAE, MAPE, and R^2 values of each test set were obtained by making five predictions for the two models, respectively, and then these evaluation indexes of the two models were compared. From Table 5, the average value of each evaluation index obtained from the five prediction results was calculated. By comparing the average values of each index of the two models, it can be seen that the improved GRU model proposed in this paper is better than the original GRU model in each evaluation index. Compared with the original GRU model, the RMSE index value of the proposed improved GRU model is reduced by 56.79%, the MAE value is decreased by 49.40%, the MAPE value is decreased by 27.91%, and the R^2 value is increased by 25.71%.

Table 5. Predictive evaluation index comparison.

Model	The Improved GRU Model				GRU Model				
	Index	RMSE	MAE	MAPE	R^2	RMSE	MAE	MAPE	R^2
Test I		314.69	440.47	0.30	0.88	748.21	889.46	0.42	0.72
Test II		348.93	467.59	0.33	0.86	774.91	908.87	0.46	0.66
Test III		302.73	423.28	0.28	0.89	768.65	895.23	0.44	0.68
Test IV		359.61	482.84	0.34	0.85	759.82	890.94	0.44	0.68
Test V		320.17	456.21	0.31	0.88	740.93	901.64	0.43	0.70
Average		329.23	454.08	0.31	0.87	758.50	897.23	0.44	0.69

The same values for training times and the number of neurons and hidden layers were used for each of the above four models, after which the training results of the respective models were contrasted. The improved GRU model in this paper is compared with the original GRU model and two popular LSTM and RNN models under the same set of data.

According to Figure 10, “GRU+” represents the improved GRU model proposed in this paper. It can be seen that the prediction accuracy of the GRU model is higher than that of the RNN and LSTM models when the data sample size and training times are less, and the prediction accuracy of the proposed improved GRU model is also much higher than that of the other models.

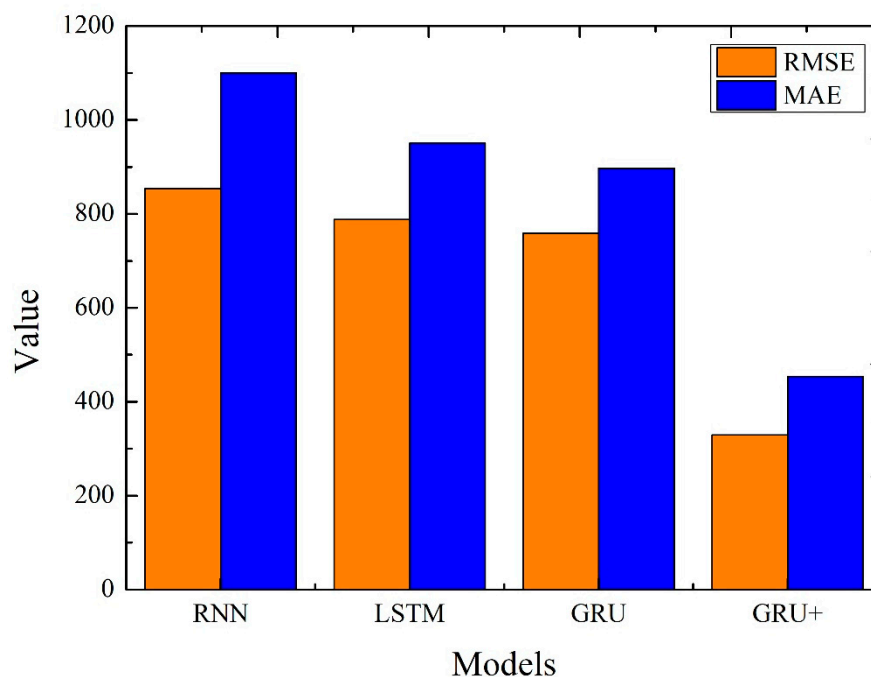


Figure 10. Comparison of RMSE and MAE values of each model.

Based on the above analysis, it can be concluded that the improved GRU model proposed in this paper has good advantages in short-term online ride-hailing travel demand prediction, and the prediction accuracy of the weighted GRU model is greatly improved compared with the ordinary GRU model.

5. Policy Implication

With the rapid change of internet travel mode, the contradiction between the rapid growth of online ride-hailing travel demand of urban residents and the supply and demand of service-vehicle resources is becoming increasingly prominent. Residents' travel behavior analysis is a very important basic work in urban comprehensive transportation-system planning and urban-construction planning, and it is also an effective basis for formulating transportation policies. Therefore, it is of great significance to study the characteristics of urban residents' travel activities. The sharing economy represented by online ride-hailing can not only directly promote urban economic development, but also stimulate relevant industries and technological innovation. Especially for cities in the period of economic transformation, if we can promote the development of new business forms with an inclusive and prudent attitude and make use of the shared economy to revitalize social resources, it will create good opportunities for urban leapfrog development. The research on online ride-hailing travel demand prediction can improve the Government's understanding of urban travel characteristics. At the same time, this research is of great significance for reducing environmental pollution, reducing urban road traffic pressure, and for urban sustainable development. The online ride-hailing travel demand forecast studied in this paper can give reasonable data for the supply and demand in different periods of time. Using the predicted results for reference, social network ride-hailing resources can be better allocated. When residents need less travel, it can reduce the number of times they use online ride-hailing and reduce road traffic pressure and environmental pollution.

6. Conclusions

The research in this paper has significance for optimizing the supply relationship of on-line ride-hailing, reducing urban road traffic pressure, and reducing traffic pollution. Based on the predicted travel demand for online ride-hailing in the future, we can reasonably arrange the number of online ride-hailing opportunities, reduce unnecessary vehicle waste, and reduce traffic pollution, which is of great significance to the sustainable development of the city.

This paper fills the gap in online ride-hailing travel demand prediction considering multiple factors. Based on the big-data analysis of online ride-hailing travel based on the SQL database, the time weight is reasonably set, and the impact of rainfall intensity on travel demand is considered at the same time. Combining the rainfall intensity and time travel characteristics with the deep learning GRU model, and analyzing the prediction results through various evaluation indexes, we can show that the prediction accuracy of the GRU model considering multiple factors proposed in this paper is better than earlier models.

Additionally, there are some future developments proposed in this article:

- (1) Weight parameters can be set more accurately through big-data analysis.
- (2) In this paper, the travel characteristics of morning and evening peak hours and late-night hours have not been taken into account. These two factors can be taken into account in the weight-setting in the future.
- (3) In the future, geographic coordinates, such as setting the weight according to the travel demand of the travel area, can also be taken into account in the model.

Author Contributions: Conceptualization, Q.Q. and R.C.; methodology, Q.Q.; software, H.G.; validation, Q.Q., H.G. and R.C.; data curation, Q.Q.; writing—original draft preparation, Q.Q.; writing—review and editing, Q.Q.; visualization, H.G.; supervision, R.C.; project administration, R.C.; funding acquisition, H.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Program of Humanities and Social Science of Education Ministry of China (Grant No. 20YJA630008) and the Ningbo Natural Science Foundation of China (Grant No. 202003N4142), the Natural Science Foundation of Zhejiang Province, China (Grant Nos. LY22G010001, LY20G010004), and the K.C. Wong Magna Fund in Ningbo University, China.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Restrictions apply to the availability of these data. Data was obtained from “Didi Chuxing” and are available from the “<https://gaia.didichuxing.com>” with the permission of “Didi Chuxing”.

Acknowledgments: This work is supported by the Program of Humanities and Social Science of Education Ministry of China (Grant No. 20YJA630008) and the Ningbo Natural Science Foundation of China (Grant No. 202003N4142), the Natural Science Foundation of Zhejiang Province, China (Grant Nos. LY22G010001, LY20G010004), and the K.C. Wong Magna Fund in Ningbo University, China.

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

References

1. Jiang, W.; Zhang, H.; Long, Y.; Chen, J.; Sui, Y.; Song, X.; Shibasaki, R.; Yu, Q. GPS data in urban online ride-hailing: The technical potential analysis of demand prediction model. *J. Clean. Prod.* **2021**, *279*, 123706. [CrossRef]
2. Ke, J.; Feng, S.; Zhu, Z.; Yang, H.; Ye, J. Joint predictions of multi-modal ride-hailing demands: A deep multi-task multi-graph learning-based approach. *Transp. Res. Part C-Emerg. Technol.* **2021**, *127*, 103063. [CrossRef]
3. Rahman, M.H.; Rifaat, S.M. Using spatio-temporal deep learning for forecasting demand and supply-demand gap in ride-hailing system with anonymized spatial adjacency information. *IET Intell. Transp. Syst.* **2021**, *15*, 941–957. [CrossRef]
4. Zhang, D.; Xiao, F.; Shen, M.; Zhong, S. DNEAT: A novel dynamic node-edge attention network for origin-destination demand prediction. *Transp. Res. Part C-Emerg. Technol.* **2021**, *122*, 102851. [CrossRef]
5. Elman, J.L. Distributed representations, simple recurrent networks, and grammatical structure. *Mach. Learn.* **1991**, *7*, 195–225. [CrossRef]

6. David, E.R.; Geoffrey, E.H.; Ronald, J.W. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533–536. [[CrossRef](#)]
7. Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Netw.* **2015**, *61*, 85–117. [[CrossRef](#)]
8. Luo, X.; Li, D.; Yang, Y.; Zhang, S. Spatiotemporal Traffic Flow Prediction with KNN and LSTM. *J. Adv. Transp.* **2019**, *2019*, 537–546. [[CrossRef](#)]
9. Yang, B.; Sun, S.; Li, J.; Lin, X.; Tian, Y. Traffic flow prediction using LSTM with feature enhancement. *Neurocomputing* **2019**, *332*, 320–327. [[CrossRef](#)]
10. Lu, X.J.; Ma, C.X.; Qiao, Y.H. Short-term demand forecasting for online car-hailing using ConvLSTM networks. *Phys. A Stat. Mech. Its Appl.* **2021**, *570*, 125838. [[CrossRef](#)]
11. Li, Z.; Xiong, G.; Tian, Y.; Lv, Y.; Su, X. A Multi-Stream Feature Fusion Approach for Traffic Prediction. *IET Intell. Transp. Syst.* **2020**, *23*, 1456–1466. [[CrossRef](#)]
12. Wang, S.; Zhao, J.; Shao, C.; Dong, C.D.; Yin, C. Truck traffic flow prediction based on LSTM and GRU methods with sampled GPS data. *IEEE Access* **2020**, *8*, 208158–208169. [[CrossRef](#)]
13. Dai, G.W.; Ma, C.X.; Xu, X.C. Short-term traffic flow prediction method for urban road sections based on space-time analysis and GRU. *IEEE Access* **2019**, *7*, 143025–143035. [[CrossRef](#)]
14. Li, N.; Hu, L.; Deng, Z.-L.; Su, T.; Liu, J.-W. Research on GRU neural network Satellite traffic prediction based on transfer learning. *Wirel. Pers. Commun.* **2021**, *118*, 815–827. [[CrossRef](#)]
15. Ibrahim, A.T.; Hall, F.L. Effect of adverse weather conditions on speed-flow-occupancy relationships. *Transp. Res. Rec. J. Transp. Res. Board* **1994**, *1457*, 184–191.
16. Brilon, W.; Ponzlet, M. Variability of speed-flow relationships on German autobahns. *Transp. Res. Rec. J. Transp. Res. Board* **1996**, *1555*, 91–98. [[CrossRef](#)]
17. Agarwal, M.; Maze, T.H.; Souleyrette, R.R. Impacts of weather on urban freeway traffic flow characteristics and facility capacity. In Proceedings of the Mid-Continent Transportation Research Symposium, Ames, IA, USA, 18–19 August 2005; Volume 20, pp. 1121–1134. [[CrossRef](#)]
18. Zhang, D.; Kabuka, M.R. Combining weather condition data to predict traffic flow: A GRU-based deep learning approach. *IET Intell. Transp. Syst.* **2018**, *12*, 578–585. [[CrossRef](#)]
19. Li, G.F.; Yang, Y.F.; Qu, X.D. Deep learning approaches on pedestrian detection in hazy weather. *IEEE Trans. Ind. Electron.* **2020**, *67*, 8889–8899. [[CrossRef](#)]
20. Liu, L.J.; Chen, R.C. A novel passenger flow prediction model using deep learning methods. *Transp. Res. Part C Emerg. Technol.* **2017**, *84*, 74–91. [[CrossRef](#)]
21. Nejadettehad, A.; Mahini, H.; Bahrak, B. Short-term demand forecasting for online car-hailing services using Recurrent Neural Networks. *Appl. Artif. Intell.* **2020**, *34*, 674–689. [[CrossRef](#)]
22. Jin, Y.; Ye, X.; Ye, Q.; Wang, T.; Cheng, J.; Yan, X. Demand forecasting of online car-hailing with Stacking Ensemble Learning approach and large-scale datasets. *IEEE Access* **2020**, *8*, 199513–199522. [[CrossRef](#)]
23. Tian, Y.; Wu, Q.; Zhang, Y. A Convolutional Long Short-Term Memory Neural Network Based Prediction Model. *Int. J. Comput. Commun. Control* **2020**, *15*, 3906. [[CrossRef](#)]
24. Li, H.; Wang, J.; Ren, Y.; Mao, F. Intercity online car-hailing travel demand prediction via a Spatiotemporal Transformer Method. *Appl. Sci.* **2021**, *11*, 11750. [[CrossRef](#)]
25. Zheng, Y.; Xue, M.; Li, B.; Chen, J.; Tao, Z. Spatial characteristics of extreme rainfall over China with hourly through 24-h accumulation periods based on national-level hourly rain gauge data. *Adv. Atmos. Sci.* **2016**, *33*, 1218–1232. [[CrossRef](#)]
26. Moayedi, H.; Osouli, A.; Nguyen, H.; Rashid, A.S.A. A novel Harris hawks’ optimization and k-fold cross-validation predicting slope stability. *Eng. Comput.* **2021**, *37*, 369–379. [[CrossRef](#)]
27. Vabalas, A.; Gowen, E.; Poliakoff, E.; Casson, A. Machine learning algorithm validation with a limited sample size. *PLoS ONE* **2019**, *14*, e0224365. [[CrossRef](#)]
28. Xiong, Z.; Cui, Y.; Liu, Z.; Zhao, Y.; Hu, M.; Hu, J. Evaluating explorative prediction power of machine learning algorithms for materials discovery using k-fold forward cross-validation. *Comput. Mater. Sci.* **2020**, *171*, 109203. [[CrossRef](#)]
29. Wu, W.; Liu, R.; Jin, W.; Ma, C. Stochastic bus schedule coordination considering demand assignment and rerouting of passengers. *Transp. Res. B Methodol.* **2019**, *121*, 275–303. [[CrossRef](#)]
30. Cheng, R.J.; Ge, H.X.; Wang, J.F. An extended continuum model accounting for the driver’s timid and aggressive attributions. *Phys. Lett. A* **2017**, *381*, 1302–1312. [[CrossRef](#)]
31. Sun, Y.Q.; Ge, H.X.; Cheng, R.J. An extended car-following model considering driver’s memory and average speed of preceding vehicles with control strategy. *Phys. A Stat. Mech. Its Appl.* **2019**, *521*, 752–761. [[CrossRef](#)]
32. Jiang, C.T.; Ge, H.X.; Cheng, R.J. Mean-field flow difference model with consideration of on-ramp and off-ramp. *Phys. A Stat. Mech. Appl.* **2019**, *513*, 465–467. [[CrossRef](#)]