Research Article

Multistep Prediction of Bus Arrival Time with the Recurrent Neural Network

Zhi-Ying Xie,^{1,2} Yuan-Rong He,^{1,2} Chih-Cheng Chen ,^{3,4} Qing-Quan Li,⁵
and Chia-Chun Wu ⁶

¹School of Computer and Information Engineering, Xiamen University of Technology, Xiamen 361024, China

²Digital Fujian Institute of Natural Disaster Monitoring Big Data, Xiamen, Fujian 361024, China

³Department of Automatic Control Engineering, Feng Chia University, Taichung 40724, Taiwan

⁴Department of Aeronautical Engineering, Chaoyang University of Technology, Taichung 413, Taiwan

⁵Shenzhen Key Laboratory of Spatial Smart Sensing and Services, Shenzhen University, Shenzhen 518060, China

⁶Department of Industrial Engineering and Management, National Quemoy University, Kinmen 892, Taiwan

Correspondence should be addressed to Chih-Cheng Chen; ccc@gm.cyut.edu.tw and Chia-Chun Wu; ccwu0918@nqu.edu.tw

Received 30 October 2020; Revised 3 January 2021; Accepted 18 February 2021; Published 12 March 2021

Academic Editor: Bosheng Song

Copyright © 2021 Zhi-Ying Xie et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Accurate predictions of bus arrival times help passengers arrange their trips easily and flexibly and improve travel efficiency. Thus, it is important to manage and schedule the arrival times of buses for the efficient deployment of buses and to ease traffic congestion, which improves the service quality of the public transport system. However, due to many variables disturbing the scheduled transportation, accurate prediction is challenging. For accurate prediction of the arrival time of a bus, this research adopted a recurrent neural network (RNN). For the prediction, the variables affecting the bus arrival time were investigated from the data set containing the route, a driver, weather, and the schedule. Then, a stacked multilayer RNN model was created with the variables that were categorized into four groups. The RNN model with a separate multi-input and spatiotemporal sequence model was applied to the data of the arrival and leaving times of a bus from all of a Shandong Linyi bus route. The result of the model simulation revealed that the convolutional long short-term memory (ConvLSTM) model showed the highest accuracy among the tested models. The propagation of error and the number of prediction steps influenced the prediction accuracy.

1. Introduction

The rapid and continuous development of China has led to an increase in the number of vehicles. The National Bureau of Statistics of China announced that the number of privately owned vehicles reached 261.5 million in 2019 with 21.22 million vehicles increased in a year. 96 cities in China had more than one million registered vehicles [1]. The rapid increase of vehicles causes traffic congestion, parking problems, and environmental pollution. Public transportation affords a larger number of passengers and alleviates such problems. Mass transportation consumes less energy and emits less amount of pollutants than private transport. Therefore, urban planning puts a priority on public

transportation. New technologies such as bus rapid transit (BRT) and driverless bus have been developed significantly with huge investment to support the public transportation system. However, a trip by bus takes a relatively long time and is not punctual, which makes people avoid it. Encouraging people to use buses more often requires optimized bus routes and punctuality of bus operation [2, 3]. However, the absence of an accurate operation schedule often causes long waiting times and bus bunching on the same route. For the punctual operation of the public buses, the bus schedule needs to be optimized, which needs an accurate prediction of the arrival time of buses on a route accurately. This not only meets the demand of ordinary passengers who want to know the arrival times of a bus at boarding stations but also optimizes the

intelligent bus scheduling system and improves the operation efficiency of the bus company.

Several neural networks have been used to predict the arrival time of a bus: non-RNN network, RNN with the time series, and temporal and spatial RNN network. Several studies adopted non-RNN networks for predicting bus arrival and operation times using (1) MapReduce-based clustering with K -means [4], (2) a backpropagation (BP) neural network model [5], (3) a particle swarm algorithm [6], (4) a wide-depth recursive (WDR) learning model [7], and (5) RNN with the time series such as long short-term memory (LSTM) [8]. Models with LSTM processed the historical data of the global position system (GPS) and bus stop locations with the influence of different routes, drivers, weather conditions, time distribution [9], heterogeneous traffic flow, and real-time data [10–12]. The temporal and spatial RNN network with ConvLSTM or a spatiotemporal property model (STPM) was originally used to predict the precipitation [13]. However, it was also used for predicting bus arrival times based on the total operation time of a bus on a route, waiting and on-board times, transfer location wait times [14–16], and multilane short-term traffic flow [17] and for creating the multistep deep neural network [18].

The bus is running on fixed lines with fixed stations. The spatial relationship between its stations determines the arrival times in the time series. Thus, this study used an RNN to predict the arrival time of a bus. A route of a bus has 30–40 bus stations in general. Arrival time prediction includes the time prediction of each station along the way from the starting to the finishing stop, the arrival times at subsequent stations, and the arrival time of the nearest vehicle to a station. This study first analyzed the bus arrival time. Based on the analysis, the input eigenvectors of a neural network were defined, and then, seven RNN models for predicting the arrival time from four categories were tested. Then, the proposed model was trained by the measured data of arrival and departure times of the buses in a route of Linyi, Shandong Province. Then, the multistep prediction of the arrival time was carried out.

This paper is organized as follows. Section 2 describes the theoretical background and introduces the recurrent neural network. Section 3 describes the pretreatment and analysis of data. Section 4 discusses the analysis result of the RNN model. Finally, Section 5 concludes this study.

2. Theoretical Background

A recurrent neural network (RNN) [19] has a feedback structure that processes sequential data for time-series prediction or classification. RNN is widely used in various applications, and new models using it have been suggested such as LSTM, GRU, and ConvLSTM. According to the data in this study, we divided the prediction into four categories and adopted a multistep prediction for bus arrival times. The time-series input data is essential for the prediction with optimal feature extraction and memory efficiency. The data is processed in an RNN with internal feedback and feed-forward connection, which retain and reflect the state or memory of a long context window [20]. The RNN suffers from a common disadvantage of the gradient disappearance

(gradient vanishing) and gradient explosion problem [21–23], which results in limited applications due to training problems. To solve the problems, Hochreiter et al. [24] proposed and continued improving LSTM for different applications [25, 26]. LSTM specializes in memorizing long sequences and effectively avoiding the problem of gradient disappearance. Hidden layers of LSTM use memory blocks that store the previous sequence information, while increasing the performance of three gates: input, output, and forget gates. These control the sequence information for memory. The gated recurrent unit (GRU) [27] is a modestly simplified LSTM. GRU combines the forget and input gate into an update gate and the cell and hidden state. A model with GRU is simpler and has less activation function and output computation than the standard LSTM model.

2.1. Pure LSTM and Pure GRU Model. Figure 1 shows the hidden units of LSTM which are replaced by memory blocks.

Calculating c_t and h_t requires the following equations:

$$\begin{aligned} i_t &= \sigma(W_i X_t + U_i h_{t-1} + b_i) \text{ (Input gate),} \\ f_t &= \sigma(W_f X_t + U_f h_{t-1} + b_f) \text{ (Forget gate),} \\ o_t &= \sigma(W_o X_t + U_o h_{t-1} + b_o) \text{ (Output gate),} \\ \tilde{c}_t &= \tanh(W_c X_t + U_c h_{t-1} + b_c) \text{ (New memory cell),} \\ c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \text{ (Memory cell),} \\ h_t &= o_t \odot \tanh(c_t). \end{aligned} \quad (1)$$

In these equations, $\odot$ Hadamard product is the multiplication of the corresponding elements in the operation matrix, W_i , W_f , W_o , and W_c are the weights of X_t , U_i , U_f , U_o , and U_c are the weights of h_{t-1} , b_i , b_f , b_o , and b_c are the bias conditions, σ is the sigmoid function, and $\tanh$ is the hyperbolic tangent function.

Figure 2 shows the GRU. There is only one hidden state h_t in GRU. Through the linear transformation of the input tensor and hidden state, the weighted sum of the hidden state inflow is calculated with equations (2) and (3). The linear transformation for r_t , h_{t-1} , and the input tensor is combined with the activation function of equation (4) to calculate the updated value of the hidden state. The mixed weight for calculation of the implicit state in the previous step is shown in equation (5). The final output h_t is the same as LSTM. Compared with LSTM, there is one less activation function calculation and output calculation as well as the final hidden state update, so the calculation is relatively simple.

$$z_t = \sigma(W_z X_t + U_z h_{t-1} + b_z) \text{ (Update gate),} \quad (2)$$

$$r_t = \sigma(W_r X_t + U_r h_{t-1} + b_r) \text{ (Reset gate),} \quad (3)$$

$$\tilde{h}_t = \tanh(W_h X_t + r_t \odot U_h h_{t-1} + b_h) \text{ (New memory cell),} \quad (4)$$

$$h_t = (1 - z_t) \odot \tilde{h}_t + z_t \odot h_{t-1}. \quad (5)$$

FIGURE 1: LSTM memory cell structure.

FIGURE 2: GRU unit structure.

Figure 3 illustrates a pure network model of LSTM and GRU. The input layer has the sequence of the arrival time series input, and the other two layers use a fully connected prediction network. Input and output data are 3D tensors with a shape $[?, 41, 1]$ ("?" means that a dimension can have any length).

These models use a single layer of LSTM and GRU. In the input layer, variables are, such as route, direction, vehicle model, and driver, also regarded as a part of the time sequence.

2.2. Multi-Input Model Separated by Time Series. As the variable is not sensitive to any specific ordering, the RNN cannot process it alone. However, a BP network can process through a connection layer. Thus, the integration of RNN and BP was used for the prediction network (Figure 4).

The integrated network was in accordance with the characteristics of the input data. A two-part network used the time series-related input data such as route number, driver, departure time, and route length for LSTM processing. Through a connection layer, the prediction layer was processed. Since time series input data became shorter even with the addition of LSTM, the total trainable parameters were not significantly increased compared with pure LSTM.

2.3. LSTM Stacking Model. To achieve better accuracy of the prediction than a single layer, a multilayer LSTM was

employed. Stacking four LSTMs had hidden units in 256, 128, 64, and 32 layers, respectively. Figure 5(a) shows the diagram of the stacking models. There is also a two-way LSTM composition, in which the forward and backward connections also employ a reverse projection function, which is suitable in our case to verify arrival time predictions. Figure 5(b) shows the diagram of the bidirectional network models.

2.4. Spatiotemporal Time-Series Model. The bus operation is in a space-time domain although there are little changes in the spatial dimension for an operation in the fixed route. As ConvLSTM processes the data of time and space, it integrates a convolution of time and space into calculating each gate of LSTM. The following equations are used for the calculation:

$$i_t = \sigma(W_i * X_t + U_i * h_{t-1} + b_i) \text{ (Input gate),} \quad (6)$$

$$f_t = \sigma(W_f * X_t + U_f * h_{t-1} + b_f) \text{ (Forget gate),} \quad (7)$$

$$o_t = \sigma(W_o * X_t + U_o * h_{t-1} + b_o) \text{ (Output gate),} \quad (8)$$

$$\tilde{c}_t = \tanh(W_c * X_t + U_c * h_{t-1} + b_c) \text{ (New memory cell),} \quad (9)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \text{ (Memory cell),} \quad (10)$$

$$h_t = o_t \odot \tanh(c_t). \quad (11)$$

A ConvLSTM network with batch normalization (BN) consists of a specification and flattening layer and a prediction network. Figure 6 shows the diagram of the network that has a long training time and many parameters in more than five dimensions. This network is appropriate to process time-series data with spatial properties such as bus arrival times with high accuracy.

3. Pretreatment and Analysis of Data

3.1. Data Characteristics. A bus was equipped with a device that included a GPS and data communication module. The device transmitted data to a bus scheduling system. Table 1 shows the data structure of the reporting system.

The data of arrival and departure of a bus at a bus stop consists of route, speed, arrival and departure time, coordination, and driver's number. For obtaining the Lasso variable correlation [8], the bus number, number of bus stops, days in the week, distances between bus stops, arrival and departure times, and weather were included, too. The variables were grouped into two: dynamic and static variables [14]. The dynamic variables include driving times between bus stops, staying times, and weather, while the static variables include a route, direction, vehicle model, driver, arrival and departure times, days of the week, holidays, and working days. We selected variables related to the route and the arrival times at the previous stops as the input

FIGURE 3: A pure LSTM and pure GRU network model.

FIGURE 4: An LSTM-BP integrated network model.

features to predict the arrival time at a bus stop. Input and output features are as follows:

Input feature: [route_no, direction, bus_no, driver_no, departure hours, departure minutes, days of the week, holidays, distance, and weather ($x_{t-k}, \dots, x_t$)]

Output prediction series: ($-x_{t-k}, \dots, 0, x_{t+1}, \dots, x_{t+n}$)

where x_t is the difference of the arrival time between the current station and the previous station.

3.2. Data Preprocessing

Step 1. Generating a Sample Dataset

According to the data in Section 3.1, an arrival time series was obtained from the data of arrival and departure times of a bus at a bus stop. For the convenience of calculation, the difference of the arrival times between two bus stops was calculated in seconds. Table 2 shows the example of the dataset. The existing sequence data are 120, 220, 250, and 260 which correspond to four bus stops A, B, C, and D. This means that 120 s is needed for a bus to drive from the starting location to A, 220 s from A to B, 250 s from B to C, and 260 s from C to D. When the bus arrives at C, the prediction of the arrival time to D is only needed. The input sequence is the sequence of all arrival times from the starting location to C, and the output sequence includes 260 s from C to D and the backward sequence from C to the starting location. The length of the input sequence is shorter than that of the output sequence as it only needs to predict the time to the finishing location of a bus. When predicting the time to the finishing location, it only needs to know the sequences before it. When the bus arrives at a bus stop

FIGURE 5: (a) A LSTM stacking model. (b) A bidirectional LSTM model.

between the starting and finishing location, for a consistent sequence length, the time to the previous bus stop is input as 0.

Figure 7 shows the time-series data of real arrival times. The blue and orange line is for the input and output sequence, respectively. The sequence has the predicted times of 0 at the current bus stop. An output sequence has negative numbers to maintain the correctness of the inverted time from the starting location to the bus stop.

Step 2. Dataset Normalization

The variables had different dimensions and units which affected the results of data analysis. Thus, normalization was necessary to eliminate the differences. Standardizing with the Z-score and the minimum-maximum values were used so that the final values were ranged between 0-1. The equation for standardization is as follows:

$$x_{\text{norm}} = \frac{x - \min(x)}{\max(x) - \min(x)}. \quad (12)$$

FIGURE 6: ConvLSTM network model.

FIGURE 7: A diagram of the input and output sequence generated by the time-series data of arrival times.

$$x_{\text{norm}} = \frac{x - \mu}{\sigma}, \quad (13)$$

where μ is the mean and σ is the standard deviation of the data.

In this paper, after sorting the vehicle number and driver number on the route, the sorted sequence was used as the input data and normalized by equation (7). The route number, route direction, departure time (hh:mm), days of the week, holidays, distances from the starting location, weather, and other information were normalized, and their arrival time series were processed by equation (8).

4. Analysis Result of the RNN Model

4.1. Dataset. The experiment was based on the data collected from March 28 to June 28, 2020, in Linyi, Shandong Province. The data was obtained from buses that ran on route no. 30 which had 36 bus stops (Figure 8). Tensorflow-GPU 2.0 was used for data processing and algorithm creation. The numbers of the dataset were 122,336 after pre-treatment, 78,303 in the training set, 19,590 in the verification set, and 24,443 in the test set.

4.2. Training the RNN Prediction Model. Seven network models were designed, trained, verified, and tested by using the preprocessed dataset. Pure LSTM and GRU were RNN models. LSTM-BP and GRU-BP were multiple input models with variable features separated from the time series. Bi-directional LSTM (LSTM-Bi) and LTSM-Stack were LSTM stack models. ConvLSTM was a spatiotemporal sequence model. Table 3 shows a model structure and a comparison of the parameters of the RNN network. Pure GRU had the smallest number of the parameter, while the stack model had the largest number.

The loss function selected the average absolute error (MAE) which was the difference between the prediction and real value. All network parameters were updated using the Adam optimization algorithm. The Adam algorithm performs first-order optimization. The first- and second-order optimizations were used for a dynamic design of independent adaptive learning rates for different parameters. The

TABLE 1: The data structure from a bus.

No.	Name	Types	Remarks
1	Route_no	Long	Route number
2	Route_name	String	Route name
3	Route_subno	Long	Run method no.
4	Up down	Int	Direction on route
5	Bus_no	Long	Vehicle number
6	Speed	Float	Speed
7	datetime_in	String	Entry time
8	driver_no	Long	Driver number
9	busstop_no	Long	Site number
10	busstop_name	String	Site name
11	busstop_lng	Double	Site longitude
12	busstop_lat	Double	Site latitude
13	busstop_serial	Int	Site serial number
14	busstop_type	Int	Site type
15	pack_datetime	String	Outbound time
16	inform_type	Int	Inform mode
17	netpack_type	Int	Entry mode

TABLE 2: Examples of the arrival time-series dataset format.

Input sequence	Output sequence
(120, 220, and 250)	(-120, -220, 0, and 260)
(120, 220, and 0)	(-120, 0, 250, and 260)
(120, 0, and 0)	(0, 220, 250, and 260)

Z-score standardization uses the mean and standard deviation of the data and is calculated as follows:

FIGURE 8: Linyi city's 30 bus routes and site distribution.

TABLE 3: Seven RNN network model structures and their parameters.

Classification	Type of network	Type of layer	Output sequence	Number of parameters	Total number of parameters
Pure RNN	Pure GRU	GRU	(None, 41, 64)*	12864	15,009
		Dense	(None, 32)	1056	
		Dense	(None, 33)	1089	
	Pure LSTM	LSTM	(None, 64)	16896	20,065
		Dense	(None, 32)	2080	
		Dense	(None, 33)	1089	
Multi-input hybrid model	GRU-BP	Input layer	[(None, 32, 1)]	0	25,089
		Input layer	[(None, 9)]	0	
		GRU	(None, 32, 64)	12864	
		Dense	(None, 16)	160	
		GRU	(None, 32)	9408	
		Concatenate	(None, 48)	0	
		Dense	(None, 32)	1568	
	LSTM-BP	Dense	(None, 33)	1089	32,129
		Input layer	[(None, 32, 1)]	0	
		Input layer	[(None, 9)]	0	
		LSTM	(None, 32, 64)	16896	
		Dense	(None, 16)	160	
		LSTM	(None, 32)	12416	
		Concatenate	(None, 48)	0	
Stacking models	LSTM-Bi	Dense	(None, 32)	1568	78,177
		Dense	(None, 33)	1089	
		Bidirectional (LSTM (64))	(None, 41, 128)	33792	
		Bidirectional (LSTM (32))	(None, 64)	41216	
		Dense	(None, 32)	2080	
	LTSM-Stack	Dense	(None, 33)	1089	525,281
		LSTM	(None, 41, 256)	264192	
		LSTM	(None, 41, 128)	197120	
		LSTM	(None, 41, 64)	49408	
		LSTM	(None, 32)	12416	
		Dense	(None, 32)	1056	
		Dense	(None, 33)	1089	

TABLE 3: Continued.

Classification	Type of network	Type of layer	Output sequence	Number of parameters	Total number of parameters
Space-time model	ConvLSTM	ConvLSTM2D	(None, 41, 1, 1, 41)	62156	117,233
		BN	(None, 41, 1, 1, 41)	164	
		Flatten	(None, 1681)	0	
		Dense	(None, 32)	53824	
		Dense	(None, 33)	1089

FIGURE 9: Loss function and accuracy diagram during training.

estimations were better than the traditional gradient descent method. Since the output was a time series, cosine similarity was used to determine the accuracy.

Figure 9 shows the training loss and accuracy when an epoch is 1000, a batch is 100, and a verification set is 20% of the training set. As the seven models show similar MAEs, only the training output of the ConvLSTM model was selected. In the process of training, the trend of training and validation data is consistent, and there is no fitting case. The super parameters of the selected model are suitable, too. As seen from Figure 10, the pure GRU had fewer training parameters and less training time than pure LSTM. The LSTM-Bi doubled the number of parameters and training time than the pure LSTM. The LSTM-Stack had five times more parameters than other models, but less training time than the ConvLSTM. The ConvLSTM had the longest training time, 5.8 times more parameters, and 12 times longer training time than the pure LSTM.

4.3. Analysis of Results. The test set was used to predict the training model, and the predicted arrival times are shown in Table 4.

The fitting degree of the real and the predicted value in Table 4 shows that the ConvLSTM provides the best prediction. The multi-input hybrid model, which separates the

parameters from the time series, not only increased the network complexity but also reduced the prediction accuracy. Table 5 shows the statistics of the prediction results by the seven models. MAE, RMSE, MAE, COS, number of training parameters, and time were used to quantitatively evaluate the seven network models. The prediction accuracy was improved from the pure LSTM to the ConvLSTM, as shown in Figure 11.

The results reveal the following:

- (1) The GRU was more efficient than the LSTM model with fewer parameters and considerable accuracy
- (2) The LSTM models except the ConvLSTM had more parameters and higher network accuracy than other models
- (3) The dataset property did not influence the results of the models but the complexity of the models
- (4) The ConvLSTM showed the highest accuracy as it processed the data of time and space, which indicated the need to include the space-related properties

In the process of arrival time-series prediction, the arrival times at subsequent bus stops were based on those at the previous bus stops. The ConvLSTM network model was selected to analyze the prediction accuracy through one- and two-step prediction and total time prediction.

FIGURE 10: Comparison of training parameters and times of seven RNN models.

TABLE 4: List of arrival times predicted by the seven models.

Network classification	Network model	Predicted arrival time series
Pure RNN	Pure LSTM	
	Pure GRU	

TABLE 4: Continued.

Network classification	Network model	Predicted arrival time series
Multi-input hybrid model	LSTM-BP
	GRU-BP

TABLE 4: Continued.

Network classification	Network model	Predicted arrival time series
Stacking models	LSTM-Bi
	LTSM-Stack
	ConvLSTM

TABLE 5: Statistics of the prediction results of seven models.

Model	RMSE	MSE	MAE	COS	Trainable parameters	Training time (s)
Pure LSTM	37.7892	1428.029	23.1300	0.9492	20,065	5120
Pure GRU	37.7575	1425.6302	23.0065	0.9494	15,009	5820
LSTM-BP	37.6818	1419.9197	22.8906	0.9498	32,129	6812
GRU-BP	37.8738	1434.4282	22.8810	0.9496	25,089	7886
LSTM-Bi	37.4615	1403.3645	22.5548	0.9507	78,177	9126
LTSM-Stack	34.6279	1199.0963	18.9767	0.9585	525,281	13172
ConvLSTM	34.3812	1182.0734	17.0876	0.9595	117,233	61773

MAE: mean absolute error; RMSE: root mean square error; COS: cosign similarity.

FIGURE 11: Prediction accuracies of the seven models.

(a)

(b)

(c)

(d)

FIGURE 12: Multistep prediction comparison chart. (a) One step. (b) Two step. (c) Total. (d) Histogram.

Figure 12 shows the test sample set on the x -axis and the difference between the predicted and real values on the y -axis. The mean and RMSE were calculated from the mean values and mean square deviation of the differences. The one-step prediction had the highest accuracy, and the total time prediction (multistep prediction) showed the lowest accuracy. The regularity in the histogram of Figure 12 reveals that the one-step prediction has the smallest deviation and the highest error, which is related to the accumulation and propagation of errors in the prediction of the arrival times of the subsequent bus stops.

5. Conclusion

The public transport system is a complex system with a high degree of uncertainty. The system is understood as a multistep prediction problem in which uncertainty leads to poor prediction accuracy. This paper first analyzed the main variables affecting this uncertainty, and then, the variables such as route, direction, vehicle, driver, departure hour, departure minute, day of the week, holiday, distance from the starting location, and weather were selected. The arrival time series before the current bus stops was also selected. These variables fully reflected the impact on the arrival time-series prediction. Among RNN networks for time-series analysis, we processed the data by using seven different network models in four different types of networks.

We analyzed and compared the predictive power of the seven RNN models with the variables and parameters in the measured dataset. We noticed an improvement in prediction accuracy by adding variables in one- and two-step prediction models, but not in the multistep (total time prediction) model. The multistep model increased the network complexity only. The ConvLSTM showed the highest prediction accuracy with spatiotemporal data. The statistics of one-, two-, and multistep prediction showed that the accumulation and propagation of the sequence prediction error caused more steps and a large deviation of the predicted time. The accurate bus arrival time prediction encourages more people to use buses for transportation and allows operating companies to optimize bus schedules for increasing the efficiency of their operation. This also improves the traffic condition in cities.

Accurate bus arrival information also relieves the anxiety of users by decreasing waiting time and helps to provide passengers with an improved service. The accurate prediction of bus arrival times can be integrated into an intelligent bus scheduling system in a smart transportation system. Such a system improves the management of a public transport system, increases the economic benefits of the system, and ultimately brings social benefits.

Data Availability

The nature of the data includes excel files, and the data can be accessed at <https://github.com/ricebow/multi-step-RNN>. There are no restrictions on data access. The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Fujian Province Natural Fund Project (Grant no. 2020J01263), Science and Technology Planning Foreign Cooperation Project of Longyan (Grant no. 2019LYF7003), Open Fund Project of Fujian University Engineering Research Center for Disaster Prevention and Mitigation of Southeast Coastal Engineering Structure of Putian University (Grant no. 2019005), and Open Foundation Project of Fujian Provincial Key Laboratory of Higher Education (Putian University) (Grant no. ST19004).

References

- [1] National Bureau of Statistics, *Statistical Bulletin of the People's Republic of China on National Economic and Social Development*, National Bureau of Statistics, Beijing, China, 2020.
- [2] H. Lu, Z. Sun, and W. Qu, "Big data and its applications in urban intelligent transportation system," *Journal of Transportation Systems Engineering and Information Technology*, vol. 15, pp. 45–52, 2015.
- [3] D. Li, Y. Yao, and Z. Shao, "Big data in smart city," *Geomatics and Information Science of Wuhan University*, vol. 39, pp. 631–640, 2017.
- [4] F. Xie, J. Gu, S. Zhang et al., "Predicting model of bus arrival time based on Map reduce clustering and neural network," *Journal of Computer Application*, vol. 37, pp. 118–129, 2017.
- [5] L. Wang, Q. Su, and R. Zheng, "Bus arrival time prediction based on Elman's dynamic neural network," *Mechanical & Electrical Technology*, vol. 35, pp. 135–139, 2012.
- [6] Y. Ji, J. Lu, X. Chen et al., "Prediction model of bus arrival time based on particle swarm optimization and wavelet neural network," *Journal of Transportation Systems Engineering and Information Technology*, vol. 16, pp. 60–66, 2016.
- [7] Z. Wang, K. Fu, and J. Ye, "Learning to estimate the travel time," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ACM, pp. 858–866, London, Kingdom, August 2018.
- [8] J. Lu, L. Sun, and Q. Shi, "Prediction of bus arrival time based on gated recurrent unit neural networks," *Journal of Nantong University (Natural Science Edition)*, vol. 19, pp. 43–49, 2020.
- [9] Q. Han, K. Liu, L. Zeng, G. He, L. Ye, and F. Li, "A bus arrival time prediction method based on position calibration and LSTM," *IEEE Access*, vol. 8, pp. 42372–42383, 2020.
- [10] A. A. Agafonov and A. S. Yumaganov, "Bus arrival time prediction using recurrent neural network with LSTM architecture," *Optical Memory and Neural Networks*, vol. 28, no. 3, pp. 222–230, 2019.
- [11] W. Xiangxue, X. Lunhui, and C. Kaixun, "Data-driven short-term forecasting for urban road network traffic based on data processing and LSTM-RNN," *Arabian Journal for Science and Engineering*, vol. 44, no. 4, pp. 3043–3060, 2019.
- [12] Z. Huang, Q. Li, F. Li, and J. Xia, "A novel bus-dispatching model based on passenger flow and arrival time prediction," *IEEE Access*, vol. 7, pp. P106453–P106465, 2019.
- [13] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W. Wong, and W. Woo, "Convolutional LSTM network," *A Machine*

- Learning Approach for Precipitation Nowcasting*, vol. 1, p. 9, 2015.
- [14] Y. Lai, L. Zhang, F. Yang, W. Lu, and T. Wang, "Bus arrival time prediction algorithm based on the spatio-temporal correlation attribute model," *Ruan Jian Xue Bao/Journal of Software*, vol. 31, no. 3, pp. 648–662, 2020.
 - [15] H. Liu, H. Xu, Y. Yan, Z. Cai, T. Sun, and W. Li, "Bus arrival time prediction based on LSTM and spatial-temporal feature vector," *IEEE Access*, vol. 8, pp. 11917–11929, 2020.
 - [16] P. He, G. Jiang, S.-K. Lam, and Y. Sun, "Learning heterogeneous traffic patterns for travel time prediction of bus journeys," *Information Sciences*, vol. 512, pp. 1394–1406, 2020.
 - [17] Y. Ma, Z. Zhang, and A. Ihler, "Multi-lane short-term traffic forecasting with convolutional LSTM network," *IEEE Access*, vol. 8, pp. 34629–34643, 2020.
 - [18] N. C. Petersen, F. Rodrigues, and F. C. Pereira, "Multi-output bus travel time prediction with convolutional LSTM neural network," *Expert Systems with Applications*, vol. 120, pp. 426–435, 2019.
 - [19] R. J. Williams and J. Peng, "An efficient gradient-based algorithm for on-line training of recurrent network trajectories," *Neural Computation*, vol. 2, no. 4, pp. 490–501, 1990.
 - [20] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A critical review of recurrent neural networks for sequence learning," 2015, <https://arxiv.org/abs/1506.00019>.
 - [21] Lechevallier, Saporta - 2010 - in Proceedings of COMPSTAT'2010.pdf.
 - [22] T. Zhang, "Solving large scale linear prediction problems using stochastic gradient descent algorithms," *International Conference on Machine Learning. Omnipress*, vol. 116, 2004.
 - [23] K. S. Tai, R. Socher, and C. D. Manning, "Improved semantic representations from tree-structured long short-term memory networks," 2015, <https://arxiv.org/abs/1503.00075>.
 - [24] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, p. 1735, 1997.
 - [25] J. Schmidhuber and F. Cummins, "Learning to forget: continual prediction with LSTM," in *Proceedings of the 1999 Ninth International Conference on Artificial Neural Networks ICANN 99*, Edinburgh, UK, September 1999.
 - [26] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, "Learning precise timing with LSTM recurrent networks," *Journal of Machine Learning Research*, vol. 29, 2002.
 - [27] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, <https://arxiv.org/abs/1412.3555>.