

Bus Travel Speed Prediction Using Long Short-term Memory Neural Network

Seung-Bae Jeon,¹ Myeong-Hun Jeong,^{1*} Tae-Young Lee,¹
Jeong-Hwan Lee,¹ and Jae-Myoung Cho²

¹Department of Civil Engineering, Chosun University, 309 Pilmun-daero,
Dong-gu, Gwangju 61452, Republic of Korea

²Department of Civil Engineering, Songwon University, 73 Songam-ro,
Nam-gu, Gwangju 61756, Republic of Korea

(Received September 22, 2020; accepted December 1, 2020)

Keywords: long short-term memory neural network, bus travel speed prediction, digital tachograph, autoregressive integrated moving average

Improving the accuracy of public transport information has attracted attention in the development of smart cities. We aim to predict the bus travel speed on road sections using a long short-term memory (LSTM) neural network. We use digital tachograph (DTG) data combined with road link data. Motion sensors in DTG can record vehicle's operation information, such as journey distance, speed, and driving time. The experimental results show that the proposed model based on LSTM performs better than the autoregressive integrated moving average (ARIMA) model. The accuracy was improved by 20% on average.

1. Introduction

Transportation-related services have a direct impact on our lives. Commuting routes, logistics services, and delivery services are already an important part of our lives, and they can be the basis of services for a smart city. In Korea, the costs associated with traffic congestion have steadily increased from about \$22.4 billion in 2007 to \$32.2 billion in 2017.⁽¹⁾ Therefore, mitigating the traffic congestion problem can result in enormous savings to society. The revitalization of public transportation facilities is one of the factors that can address this problem. One way to encourage and promote public transport is to convey accurate public transport information to people (e.g., when the bus will arrive). If this information is not accurate, people may avoid using public transport.

However, forecasting traffic information remains a challenging task because the road conditions are dynamic and are affected by many variables (e.g., traffic lights, traffic accidents, and weather). Previous research has utilized many statistical and machine learning models to examine this problem.^(2–5)

In this study, we predicted bus travel speed using a long short-term memory (LSTM) model, which is a recurrent neural network (RNN). The experimental data was from a digital

*Corresponding author: e-mail: mhjeong@chosun.ac.kr
<https://doi.org/10.18494/SAM.2020.3111>

tachograph (DTG), which was combined with road link data. We used Amazon Elastic MapReduce (EMR), part of Amazon Web Services (AWS), as a data pre-processing process to deal with a vast amount of data. Our experimental results demonstrated that the proposed method performed better than the autoregressive integrated moving average (ARIMA) model.

In the remainder of this paper, we review previous methodologies in related works in Sect. 2. Then, we present the data and the proposed method in Sect. 3. Section 4 describes an experimental evaluation of the proposed method. Finally, we conclude with a discussion of the limitations of the approach and future work in Sect. 5.

2. Background

There have been many studies on predicting traffic information in advance, which can be largely divided into parametric and non-parametric approaches.⁽⁶⁾

Parametric methods predetermine the structure and parameters of the model. The structure of such a model is based on theoretical assumptions (e.g., a normal distribution), and model parameters can be calculated by empirical experiments. For example, the ARIMA model examines whether the time series data are stationary before applying the method.^(7,8) However, most of the data do not follow the assumption of a normal distribution in the real world and often do not have stationarity. Therefore, it is common to use differences during data pre-processing.

Non-parametric methods, unlike parametric methods, do not require basic assumptions about the data distribution and are relatively insensitive to missing data or outliers. For example, machine learning algorithms [e.g., support vector machine,⁽⁹⁾ Kalman filter model,⁽¹⁰⁾ k-nearest neighbors model⁽¹¹⁾] have been applied to traffic forecasting. Furthermore, the first artificial neural network (ANN) was designed by Frank Rosenblatt of Cornell Aviation Institute in 1957, which had a multilayer perceptron to deal with nonlinear or XOR problems.⁽¹²⁾ Among the ANNs, densely connected networks and convnets have no memory. Each input is handled independently of the others. These networks are called feedforward networks. However, they have difficulty dealing with sequence or temporal series data. The entire sequence must be handled in the network, rather than each data point being treated independently, to process the data of a sequence or temporal series. The RNN was therefore developed to process a sequence by retaining each element of the sequence's inclusion information.⁽¹³⁾

However, the main problem of traditional RNNs is that it is impossible for them to learn long-term dependences. Training feedforward neural networks such as RNNs generates a vanishing gradient problem and they cannot be trained to connect an entire sequence of data. LSTM has been proposed to address these limitations of traditional RNNs.⁽¹⁴⁾ LSTM can solve the vanishing gradient problem and has been used in traffic prediction.⁽¹⁵⁾

3. Materials and Method

The experimental data in the study consisted of a combination of DTG and road link data. The combined data were used to predict the speed of buses on road sections. The routes of the

No. 30-3 Hanam bus in Seoul were used in this study. The Gyeonggi Sangwoon Corporation operates the buses, and the total number of buses is between 18 and 26 per day.

3.1 Road link data

The operating time of the No. 30-3 bus is from 4:15 a.m. to 11:40 p.m., with a bus running every 12 min. There are 55 stops along the bus route, which has a total length of approximately 15 km. It is necessary to link DTG data to road link data to predict the speed travelled along particular road sections. The initial road link data were composed of 144 links. We simplified the road link data by reducing the route to 13 links to combine the data with DTG data. Figure 1 presents the bus route, showing the 13 road links.

3.2 DTG data

The records of DTG data are about 300 million rows (approximately 800 GB), dating from September 1 to 30, 2016. We developed a Pig script based on Hadoop for cleaning the data to extract the No. 30-3 bus route. The Hadoop software library is a framework that enables the distributed processing of large data sets on multiple computer clusters using a simple programming model.

To combine the DTG data with the road link data, we assigned each DTG point to a unique road link ID by considering the nearest road link. However, the inherited GPS errors gave rise to difficulties in combining DTG data with road link data. Therefore, we used the azimuth angle to assign the correct road link ID to the DTG data. Figures 2 and 3 show the assignment



Fig. 1. (Color online) Road link data. Different colors represent different road links.

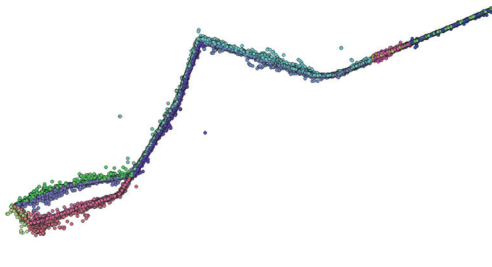


Fig. 2. (Color online) Assignment of nearest road link ID to DTG data.

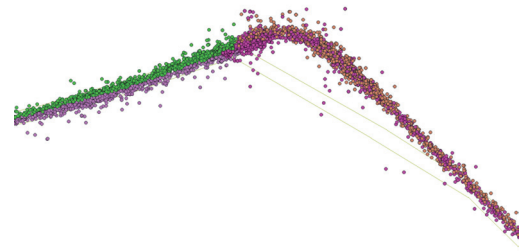


Fig. 3. (Color online) Assignment of road link ID using azimuth angle.

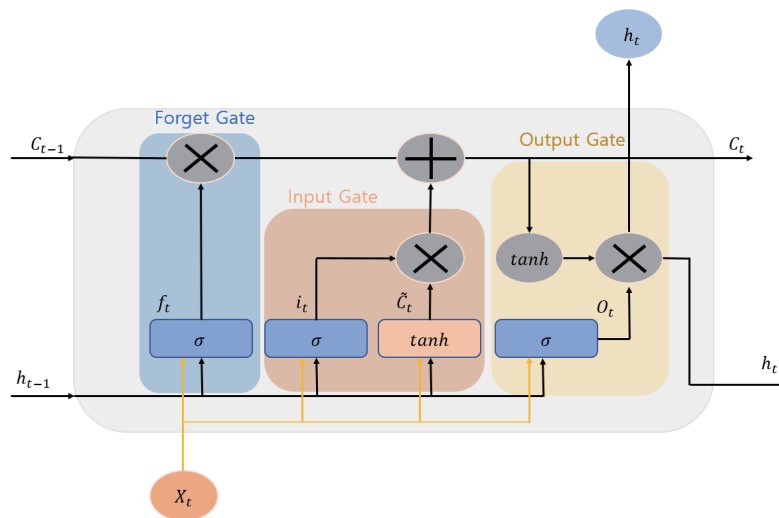


Fig. 4. (Color online) LSTM architecture based on Ref. 15.

of the road link ID to the DTG data. After combining the road link and DTG data, we prepared the experimental data sets, which show the average speed traveled on each road section at 10 min intervals.

3.3 LSTM

LSTM was proposed to solve the vanishing gradient problem of traditional RNNs. Figure 4 shows the architecture of LSTM.

LSTM can solve the long-term dependence problem of the existing RNN. Its architecture is a modification of that of the RNN. The existence of a cell state characterizes its architecture. The LSTM layer has three gates called the input, forget, and output gates.

The forget gate layer decides whether to forget, using weights for how much to forget. The state is updated by multiplying the previous state by the output of the forget gate. Also, the output values from the input gate are multiplied and added to the state. The input gate layer determines which of the new information will be contained in the cell state using the sigmoid

function. Next, the hyperbolic tangent layer creates new data candidate values. The output gate layer determines the output. The final output is calculated by multiplying by a value between -1 and 1 obtained by applying \tanh to the cell state and the sigmoid output.

We compare the proposed method with the ARIMA model, which is the most well-known model among the approaches used to predict time series. The ARIMA model assumes stationarity, which in this case means no trends and no seasonality of the underlying data. We applied the differences to the ARIMA model before the experiments.

4. Results

We divided the total data set into a training set and a test set. The training data were data for 24 days, comprising 80% of the total data, and the rest were test data. We used the DTG data ranging from 5 a.m. to 12 a.m. There were 114 data sets per day, obtained by averaging the data over 10 min intervals.

We implemented LSTM using R with Keras. R is a programming language, software environment and freeware for statistical calculations and graphics. Keras is an open source neural network library written in Python. It is designed to enable rapid experimentation with deep neural networks and focuses on the scalability of a minimal modular method. Figure 5 shows the validation graph, which helps to determine hyperparameters such as the epoch. For example, we used five epochs with 700 steps per epoch. After creating the training model, we predicted the speed of travel on the road sections, as shown in Fig. 6. These road sections are also presented using the road link ID in Fig. 7.

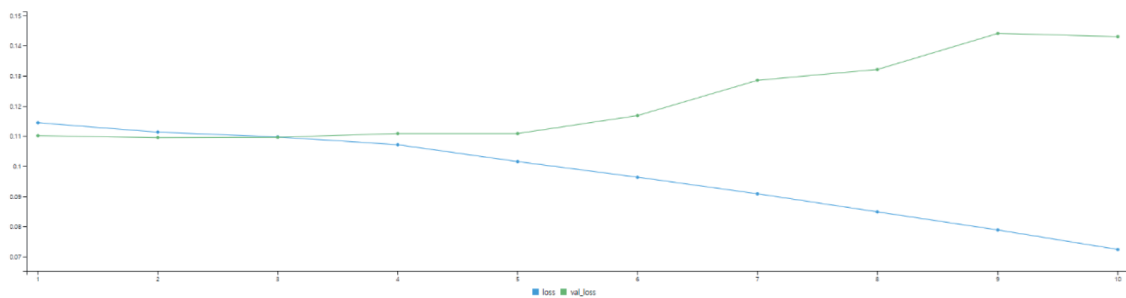


Fig. 5. (Color online) Validation graph.

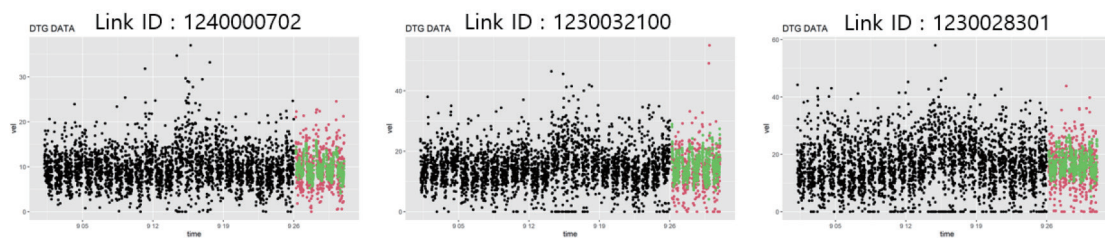


Fig. 6. (Color online) Prediction of speed using LSTM.



Fig. 7. (Color online) Road link IDs corresponding to Fig. 6.

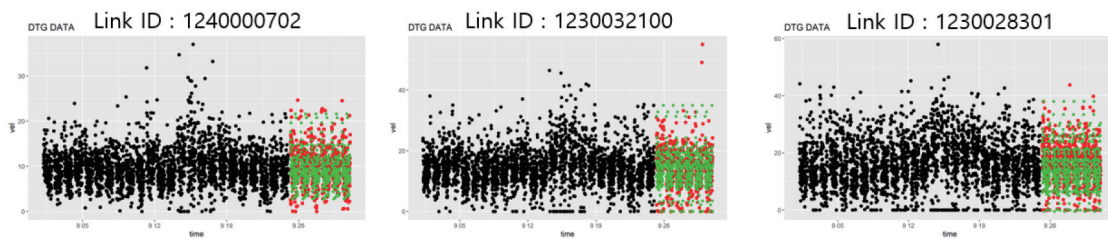


Fig. 8. (Color online) Prediction of speed using ARIMA.

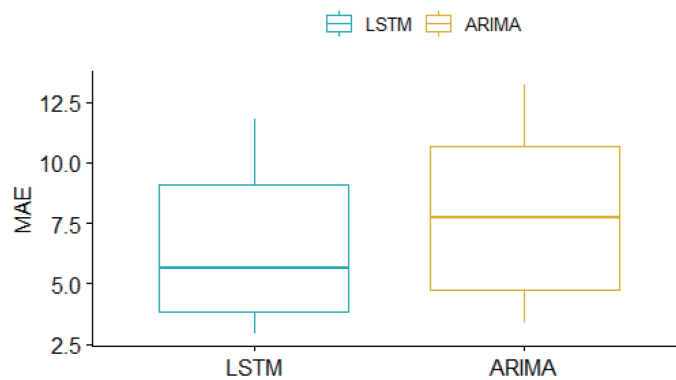


Fig. 9. (Color online) Comparison between LSTM and ARIMA using boxplot.

Black dots indicate the speeds obtained from the training data. The actual speeds of the test data are represented as red dots. Green dots indicate the predictions of the proposed method. Similarly, ARIMA was used to predict the travel speed on each road section as shown in Fig. 8.

We compared the mean absolute error (MAE) of the prediction values for the LSTM and ARIMA models. Figure 9 shows the improvement in LSTM performance over the ARIMA

model. The average improvement is approximately 20%, and the improvement for each link ranges from 5.8 to 13.6%. In addition, we conducted statistical comparisons between the two groups. The analysis of variance (ANOVA) confirmed that there was no statistically significant difference between the performance characteristics ($p = 0.338$).

5. Discussion and Conclusions

We predicted the bus travel speed on road sections using LSTM. The experimental results demonstrated that the proposed method performed better than the traditional time-series analysis method of ARIMA. The LSTM model showed an accuracy that was between 5 and 13% higher than that of the ARIMA model. However, this difference was not statistically significant.

Future work should focus on the integration of other parameters. In the current study, we only used the vehicle speed on road sections. The inclusion of additional parameters, such as the local environment or weather, may result in an improvement in the model. In addition, it is necessary to consider real-time analysis, which would provide better accessibility of public transportation.

Acknowledgments

This work was supported by a National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. 2018R1C1B5043892).

References

- 1 H.-S. Jo: 2011, 2012 Traffic Congestion Costs: Estimation and Trend Analysis, L. Ho and Y.-C. Kim, Eds. (The Korea Transport Institute, Goyang, 2014) pp. 1–121.
- 2 J. Guo and B. M. Williams: *Transp. Res. Rec.* **2175** (2010) 28. <https://doi.org/10.3141/2175-04>
- 3 H. Dong, L. Jia, X. Sun, C. Li, and Y. Qin: *Proc. 2009 5th Int. Joint Conf. INC, IMS and IDC (IEEE, 2009)* 1649–1652.
- 4 M.-H. Jeong, S.-B. Jeon, S. Park, and S. Kang: *Sens. Mater.* **31** (2019) 3827. <https://doi.org/10.18494/SAM.2019.2565>
- 5 M.-H. Jeong, J. Yin, and S. Wang: *Proc. 10th Int. Conf. Geographic Information Science (GIScience, 2018)* 6:1–6:14.
- 6 J. Van Lint and C. Van Hinsbergen: *Artif. Intell. Appl. Critical Transp. Issues* **22** (2012) 22. <https://doi.org/10.17226/22690>
- 7 D.-W. Xu, Y.-D. Wang, L.-M. Jia, Y. Qin, and H.-H. Dong: *Front Inf. Tech. Electron. Eng.* **18** (2017) 287. <https://doi.org/10.1631/FITEE.1500381>
- 8 B. M. Williams and L. A. Hoel: *J. Transp. Eng.* **129** (2003) 664. [https://doi.org/10.1061/\(ASCE\)0733-947X\(2003\)129:6\(664\)](https://doi.org/10.1061/(ASCE)0733-947X(2003)129:6(664))
- 9 L. Vanajakshi and L. R. Rilett: *Proc. 2007 IEEE Intelligent Vehicles Symposium Conf. (IEEE, 2007)* 600–605.
- 10 J. Van Lint: *IEEE Trans. Intell. Transp. Syst.* **9** (2008) 38. <https://doi.org/10.1109/TITS.2008.915649>
- 11 W. Qiao, A. Haghani, and M. Hamed: *J. Intell. Transp. Syst. Technol. Plann. Oper.* **17** (2013) 165. <https://doi.org/10.1080/15472450.2012.748555>
- 12 S. Goki: *Deep Learning from Scratch*. Eds. (O'Reilly Japan, Tokyo, 2016) Chap. 2.
- 13 F. Chollet: *Deep Learning with R*, J. J. Allaire, Eds. (Manning Publications, New York, 2018) 1st ed., Chap. 6.
- 14 S. Hochreiter and J. Schmidhuber: *Proc. 9th Int. Conf. Neural Information Processing Systems (MIT Press, 1996)* 473–479.
- 15 S. Hochreiter and J. Schmidhuber: *Neural Comput.* **9** (1997) 1735. <https://doi.org/10.1162/neco.1997.9.8.1735>