Ego-Vehicle Speed Prediction Using a Long Short-Term Memory Based Recurrent Neural Network

Article *in* International Journal of Automotive Technology · August 2019

DOI: 10.1007/s12239-019-0067-y

Kyuhwan Yeon
Hyundai Motor Company
3 PUBLICATIONS **117** CITATIONS
SEE PROFILE

Kyunghan Min
Hanyang University
26 PUBLICATIONS **327** CITATIONS
SEE PROFILE

Jaewook Shin
Hanyang University
13 PUBLICATIONS **286** CITATIONS
SEE PROFILE

Myoungho Sunwoo
Hanyang University
229 PUBLICATIONS **5,347** CITATIONS
SEE PROFILE

International Journal of Automotive Technology, Vol. 20, No. 4, pp. 713−722 (2019)
DOI 10.1007/s12239−019−0067−y

pISSN 1229−9138/ eISSN 1976−3832

EGO-VEHICLE SPEED PREDICTION USING A LONG SHORT-TERM MEMORY BASED RECURRENT NEURAL NETWORK

Kyuhwan Yeon[1)], Kyunghan Min[2)], Jaewook Shin[2)], Myoungho Sunwoo[2)] and Manbae Han[3)*]

[1)]Department of Automotive Electronics and Control Engineering, Hanyang University, Seoul 04763, Korea
[2)]Department of Automotive Engineering, Hanyang University, Seoul 04763, Korea
[3)]Department of Mechanical and Automotive Engineering, Keimyung University, Daegu 42601, Korea

(Received 8 August 2018; Revised 4 December 2018; Accepted 7 January 2019)

ABSTRACT−For predictive powertrain control, accurate prediction of vehicle speed is required. As vehicle speed prediction is affected by the driver's response to numerous driving conditions under uncertainty, the development of an accurate model is quite challenging. This paper proposes an ego-vehicle speed prediction model using a long short-term memory (LSTM) based recurrent neural network (RNN). The proposed model uses various inputs to increase the prediction accuracy: internal vehicle information, relative speed and distance to the vehicle ahead measured by a radar sensor, and the ego-vehicle location estimated by the GPS signal and B-spline roadway model. The LSTM based RNN model predicts the ego-vehicle speed for 15 seconds by using inputs from the past 30 seconds. The model was evaluated by real driving data for three scenarios: car-following, sharp curve road, and full path. In all scenarios, the radar sensor and the information of the location of the ego-vehicle contribute to improvement of the speed prediction accuracy. Thus, we conclude that for application of the predictive powertrain control, besides the internal vehicle information, the radar sensor, and the location of the ego-vehicle information are critical inputs to the speed prediction model.

KEY WORDS : Vehicle speed prediction, Recurrent neural network, Long short-term memory, Predictive powertrain control

NOMENCLATURE

x_t	: input state
h_t	: hidden state
o_t	: output state
W	: weight matrix
b	: bias matrix
i_t	: input gate
f_t	: forget gate
o_t	: output gate
C_t	: cell state
$\tilde{C}_t$	: candidate cell state
σ	: activation function
v	: average speed of 4 wheels, km/h
a	: longitudinal acceleration, m/s^2
w	: engine speed, rpm
p_b	: master cylinder pressure, bar
p_a	: accel pedal position, %
r_d	: relative distance between ego-vehicle and front vehicle, m
r_v	: relative velocity between ego-vehicle and front vehicle, km/h
s	: ego-vehicle location on the driving route
G1	: input group 1
G2	: input group 2
G3	: input group 3
X	: input states of the model
Y	: output states of the model

SUBSCRIPTS

i	: subscript for input layer
h	: subscript for hidden layer
o	: subscript for output layer
ig	: subscript for input gate
fg	: subscript for forget gate
og	: subscript for output gate
c	: subscript for cell state

1. INTRODUCTION

To reduce fuel consumption and satisfy stringent emission regulations, research on optimization of powertrain control strategies using the predicted state of the vehicle is a current focus of research. In the case of energy management strategy for hybrid electric vehicle (HEV), predicted ego-vehicle speed profiles can be used to reduce fuel consumption (Sun *et al*., 2015b; Lian *et al*., 2017). For gearshift strategies in a driving assistance system or HEV, speed prediction can also be applied to the improvement of the fuel economy (Müller *et al*., 2004; Ngo *et al*., 2011). Furthermore, predicted vehicle speed profiles can optimize

**Corresponding author.* e-mail: mbhan2002@kmu.ac.kr

cooperative adaptive cruise control (Stanger and del Re, 2013).

However, prediction of vehicle speed is difficult because it is affected by the driver's behavior in various driving environments. Route based events such as the speed limit, road width, and road curvature affect how a given driver will react (Odhams and Cole, 2004; Michael *et al*., 2016; Serna and Ruichek, 2017). Another important factor affecting vehicle speed is the movement of the vehicle ahead (Jurecki *et al*., 2017). But most of all, a driver's speed choice can vary even at the same location, which makes it even more difficult to predict the vehicle speed (Berry and Belmont, 1951).

Therefore, many studies have been conducted to solve complex and uncertain vehicle speed prediction problems (Vlahogianni *et al*., 2004, 2014; Sun *et al*., 2015a; Jiang and Fei, 2017). To date, this work has been concentrated in the field of civil engineering in order to establish a traffic management strategy (Vlahogianni *et al*., 2004, 2014). The work can be categorized by the scope and the methodology, as shown in Figure 1. In terms of the scope, there are two modeling approaches of vehicle speed prediction: the macroscopic model and the microscopic model (Bellemans *et al*., 2002). The macroscopic model forecasts traffic variables such as traffic density [vehicles/km], traffic flow [vehicles/hour], and average speed on the road. On the other hand, the microscopic model handles individual vehicle speed based on driver-behaviors (Panwai and Dia, 2007; Liu *et al*., 2011). In terms of the methodology, the vehicle speed prediction can be classified into two types: the parametric approach and the non-parametric approach (Lefèvre *et al*., 2014; Ma *et al*., 2015). The parametric approach predicts vehicle speed with a predetermined model structure based on the theoretical background. This approach uses parametric models such as the constant speed (CS) model, constant acceleration (CA) model, and intelligent driver model (Lefèvre *et al*., 2014). In the non-parametric model, the model structure and parameters are not pre-defined but are determined from the data. The non-parametric model has been studied in many papers such as the autoregressive integrated moving average model, gaussian mixture regression, and various neural networks based approaches (Ye *et al*., 2010; Park *et al*., 2011; Lefèvre *et al*., 2014).

However, in order to use predicted vehicle speed for the powertrain control rather than traffic management, A microscopic approach is preferable rather than a macroscopic approach since powertrain control is implemented in the microcontroller of the individual vehicle. Furthermore, the prediction accuracy must be guaranteed. This is because the accuracy of prediction affects predictive powertrain control performance (Sun *et al*., 2015a; Zhang *et al*., 2017). In addition, the prediction accuracy should be guaranteed not only in car-following conditions but also all driving environments because powertrain control is used regardless of specific conditions such as car-following.

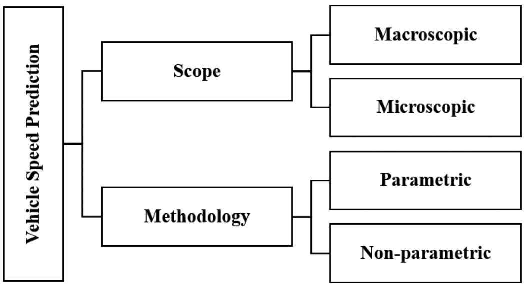


Figure 1. Category of vehicle speed prediction.

Recent studies show that non-parametric models have shown better accuracy than parametric models (Vlahogianni *et al*., 2004; Sun *et al*., 2015a). In particular, neural networks showed the best prediction result (Vlahogianni *et al*., 2004). Because the vehicle speed prediction is a complex non-linear problem, neural networks are a more effective platform for the model with non-linearity and are capable of learning the association of multivariate inputs and output patterns (Vlahogianni *et al*., 2004; Park *et al*., 2011; Guo *et al*., 2017).

One type of neural network approach uses an artificial neural network (ANN) to predict the vehicle speed with a 10 second prediction horizon (Lefèvre *et al*., 2014). Furthermore, recently, LSTM based RNNs, have drawn attention for the vehicle speed prediction (Ma *et al*., 2015; Morton *et al*., 2017). This is because LSTM based RNNs are known to improve the vanishing and exploding gradient problems (Gers *et al*., 2000). Ma *et al.* (2015) established a traffic speed prediction model based on a RNN with LSTM by using two microwave sensors on the road. This work predicts the traffic speed rather than an individual vehicle speed. Morten *et al.* designed a car-following model based on a RNN with LSTM to predict acceleration on highways. This study predicts the speed of individual vehicles only under a specific car-following condition.

Considering the studies above, these models assumed a specific driving condition. This is a clear limitation because individual vehicle speed must be accurately predicted under the various driving conditions to apply for the predictive powertrain control. To do this, not only internal vehicle data but also the relative speed and the distance with the vehicle ahead, as well as the location information of the ego-vehicle must be included in the inputs to reflect the various driving conditions.

In this paper, we propose an ego-vehicle speed prediction model using an LSTM based RNN as a microscopic and non-parametric approach to apply the various urban driving condition. Compared to previous work, our approach is not limited to car-following situations. The proposed model estimates the vehicle velocity up to a prediction horizon of 15 sec under driving conditions such as a car-following, sharp curve road and a full path combined with both conditions. To achieve this goal, the proposed model uses diverse inputs including the internal information of the ego-vehicle, the relative speed and the distance with the vehicle ahead, and the location information of the ego-vehicle. The prediction performance of the proposed model was compared to CS, CA, ANN. The proposed model was evaluated with real driving data. The influence of the inputs on the prediction accuracy was analyzed as well.

The rest of this study is organized as follows. Section 2 introduces the proposed model configuration. The method of obtaining the driving data and the driving environment are described in Section 3. The prediction results of the proposed model are analyzed in Section 4. Finally, the conclusion is presented at the end of this paper.

2. METHODOLOGY

2.1. Recurrent Neural Network

The RNN has strong prediction performance for sequential data (Lipton *et al*., 2015). Research that applies the RNN to sequential data such as time series prediction, speech recognition, and natural language is found in a number of fields (Graves *et al*., 2013; Tran *et al*., 2016; Selvin *et al*., 2017). The RNN is structured to have a feedback loop unlike a standard feedforward neural network, as shown in Figure 2. Figure 3 shows the unfolded structure of a recurrent neural network. This architecture clearly illustrates that the RNN uses an input sequence to obtain an output sequence. The RNN architecture can be described by these equations:

$$h_t = \sigma_h(W_{ih}x_t + W_{hh}h_{t-1} + b_h) \tag{1}$$

$$y_t = \sigma_0(W_{ho}h_t + b_o) \tag{2}$$

where x_t, h_t, and o_t stand for the input state, hidden state, and output state. W_{ih} is weight matrix between input layer and hidden layer, and W_{hh} is weight matrix between hidden layers (h_{t-1} and h_t). The weight matrix between the hidden layer and the output layer is represented by W_{ho} . b_h and b_o denote the bias vector to the hidden and output layer, respectively. σ_h and σ_o stand for the activation function of the hidden layer and output layer. Equation (1) shows that the hidden state is updated by input x_t and the previous hidden state, h_{t-1}. Equation (2) shows the output y_t updated by the hidden state, h_t.

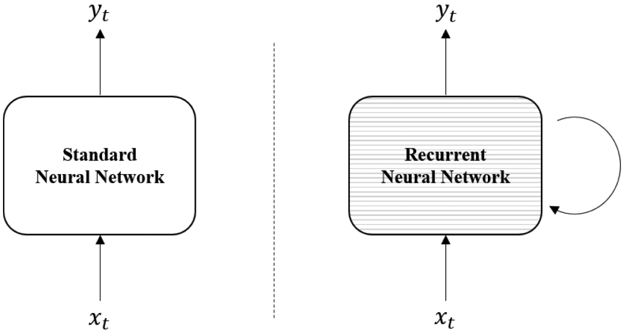


Figure 2. Standard neural network and recurrent neural network.

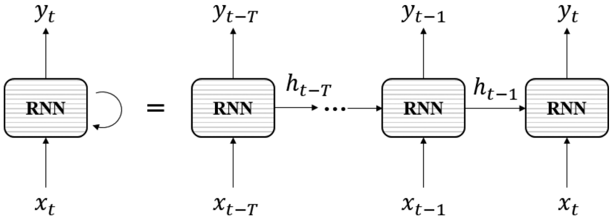


Figure 3. Unfolded structure of a recurrent neural network.

Although the RNN enables discovery of the temporal correlations between events, they have vanishing and exploding gradient problems (Pascanu *et al*., 2012). Because of these problems, the range of the input sequence should be limited. In other words, it is incapable of learning from long-term dependencies. To avoid the aforementioned disadvantages of the RNN, LSTM cell is used in this study.

2.2. LSTM Based RNN

The first study of an LSTM based RNN structure was introduced by Hochreiter and Schmidhuber (1997). A more current structure of LSTM is defined by Gers *et al*. (2000). The only difference between the traditional RNN and LSTM architecture is the hidden layer. The hidden layer of an LSTM based RNN is composed of LSTM cells. Figure 4 shows the structure of the LSTM cell. The cell can be summarized by input gate i_t, forget gate f_t, output gate o_t, cell state C_t, and hidden state h_t. These gates and states can be expressed by the following equations:

$$i_t = \sigma(W_{ig}[h_{t-1}, x_t] + b_{ig}) \tag{3}$$

$$f_t = \sigma(W_{fg}[h_{t-1}, x_t] + b_{fg}) \tag{4}$$

$$o_t = \sigma(W_{og}[h_{t-1}, x_t] + b_{og}) \tag{5}$$

$$\tilde{C}_t = \tanh(W_c[h_{t-1}, x_t] + b_c) \tag{6}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{7}$$

$$h_t = o_t \odot \tanh(C_t) \tag{8}$$

where W_{ig}, b_{ig} are weights, bias of input gate, W_{fg}, b_{fg} are weights, bias of forget gate, W_{og}, b_{og} are weights, bias of output gate, W_c, b_c are weights, bias of cell state, $\tilde{C}_t$ is the candidate cell state, σ denotes activation function, $\odot$ represents the scalar product of the two vectors. The core ideas of LSTM are the gates and the cell state. Regarding (7), the input gate determines how much of the candidate cell state is added to the new cell state, and the forget gate decides whether to keep the old cell state or not. The LSTM allows for addition or removal of information to the cell state owing to these gates. Through this process, the LSTM can handle the vanishing and exploding gradients because the cell state can effectively convey the

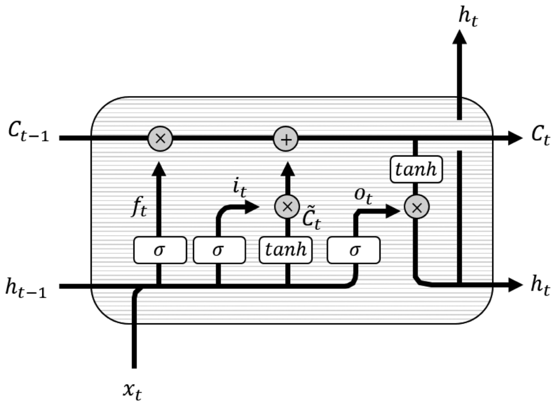


Figure 4. Architecture of LSTM. The circles are arithmetic operators, and the white boxes are the gates of LSTM.

information from the previous hidden and input states. Last output gate allows us to obtain the final result of the LSTM model.

2.3. Model Structure

We designed the ego-vehicle speed prediction model using an LSTM based RNN. Figure 5 shows the structure of the proposed model for ego-vehicle speed prediction. Input states at time *t*, *x*(*t*), is composed of combinations of three input groups which will be described in detail in the next section. *X*(*t*), final input states for the model, consists of a trace of *x*(*t*). In this study the duration of the inputs, T_{in}, was set to 30 seconds. Model output, *Y*(*t*), is a trace of predicted vehicle speed. The prediction horizon, T_{out}, is set to 15 seconds. Literature survey tells us that 15 seconds is enough to handle the energy management strategy of various applications (Lang *et al*., 2014; Rezaei and Burl, 2015a).

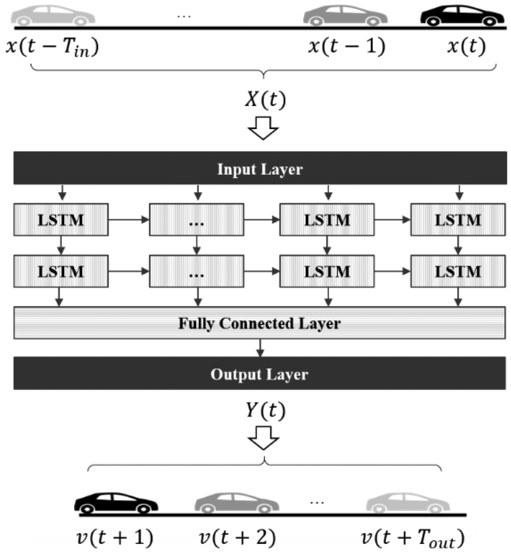


Figure 5. Structure of the proposed speed prediction model. *X*(*t*) is the input states of the model, and *Y*(*t*) is output states of the model.

Our goal in training the neural networks is finding the best weights and bias, which minimize the loss function. In this study, the root mean square error (RMSE) was used as the loss function. To find the best weights and bias, the back-propagation method is employed (Rumelhart *et al*., 1986). Furthermore, to efficiently find the optimal number of hidden states for each LSTM layer, hyperparameter optimization via radial basis function and dynamic coordinate search (HORD) was used (Ilievski *et al*., 2016). The objective of HORD is finding the best hyperparameter set, which can minimize the loss in the validation data set. The HORD algorithm changes the hyperparameters until it reaches the maximum HORD iteration number and finds the minimum validation loss. Using this algorithm, we were able to find the optimal hidden states number, which has minimum loss in the validation data.

2.4. Description of Input Groups to the Model

One of the important issues in designing a vehicle speed prediction model is selection of the input. In this paper, various inputs are used to improve the prediction performance.

Input list
① *v*: Average speed of 4 wheels
② *a*: Acceleration
③ *w*: Engine speed
④ P_{b}: Master cylinder pressure
⑤ P_{a}: Accel pedal position
⑥ r_{d}: Relative distance between the ego-vehicle and the front vehicle
⑦ r_{d}: Relative speed between the ego-vehicle and the front vehicle
⑧ *s*: Ego-vehicle location in the driving route

Input states of the model in time domain, *X*(*t*), can be written as

$$\text{G1} = \{v(t),\ a(t),\ w(t),\ p_{\text{b}}(t),\ p_{\text{a}}(t)\} \tag{9}$$

$$\text{G2} = \{r_{\text{d}}(t),\ r_{\text{v}}(t)\} \tag{10}$$

$$\text{G3} = \{s(t)\} \tag{11}$$

$$x(t) = \{\text{G1, G2, G3}\} \tag{12}$$

$$X(t) = [x(t),\ x(t\text{–}1),\ \dots\ x(t - T_{\text{in}})] \tag{13}$$

Model output state, *Y*(*t*), can be described as:

$$Y(t) = [v(t+1),\ v(t+2),\ \dots\ v(t + T_{\text{out}})] \tag{14}$$

As briefly mentioned in the earlier section, *x*(*t*), input states at the time, is composed of combinations of input groups, specifically Group 1, Group 2, and Group 3. Group 1 represents the internal vehicle information obtained from the on-board sensors in the vehicle. Inputs from G1 are related to represent vehicle speed and indicate the driver's intent to the vehicle speed. Group 2 contains inputs related to the influence of the vehicle ahead. The speed of a car is dominantly influenced by the speed of the front car in car-following situations research (Blanch Micó *et al.*, 2017; Jurecki *et al.*, 2017). Owing to this fact, the distance and speed from the front vehicle were used as inputs through measured signals from the radar sensor. Group 3 is an input related to the route-based events that affects the vehicle speed. The driver's responses to the environment of the driving route, such as the width of the road, the curvature of roads, the speed limit, and the speed bump are important factors affecting the vehicle speed (Müller *et al.*, 2004; Keulen *et al.*, 2009; Rezaei and Burl, 2015b; Michael *et al.*, 2016; Lim *et al.*, 2017). However, these events are difficult to be integrated into the inputs because these are hard to calculate arithmetically, and their boundaries are ambiguous. To avoid this problem, the location of the ego-vehicle in the route was used because the driver's response to route-based events takes place based on the location. Based on this fact, we select the location information as one of the inputs, G3.

3. REAL DRIVING ENVIRONMENT

3.1. Driving Environment

The 5.1 km driving route includes urban roads and a narrow one-way road in Seoul, Korea. Figure 6 shows the top view of the driving route. The dotted blue line is the narrow one-way road, the black line is the urban road, and the green circle is the location of the sharp curve. Figure 7 shows the vehicle speed profiles on the route, indicating that the speed profiles are affected by the location of the ego-vehicle on the driving route, especially on the curve and narrow road. On the narrow one-way road, low speed is observed because the width of the lane is so narrow. The curvature on the driving route is shown in Figure 8. The black line has a normal curvature, and two green peaks on the graph represent the sharp curve on the road. On the sharp curve, the speed is reduced because the driver cannot withstand lateral acceleration from the curve, as shown in Figure 7.

We drove this route 46 times. The driving data was divided into three parts. 34 laps were for the model training and 6 laps for the validation used in hyperparameter optimization. The remained 6 laps were for the test of the model and its performance evaluation.

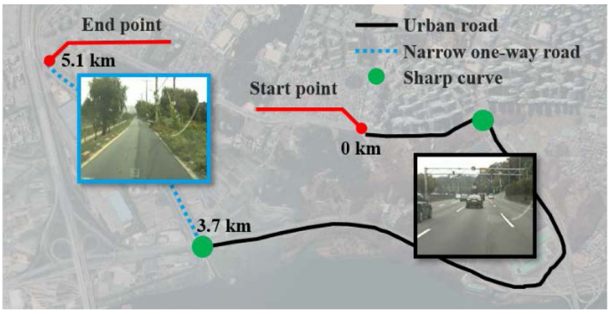


Figure 6. Top view of the driving route.

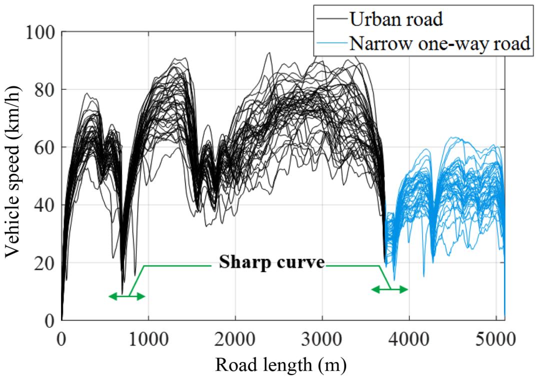


Figure 7. Vehicle speed profiles on the driving route.

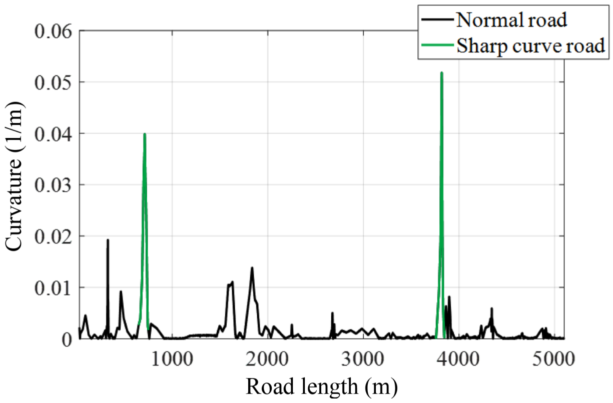


Figure 8. Curvature of the driving route.

3.2. Data Acquisition

To log the data for the modeling, a Kia Sorento equipped with radar and RTK-GPS (Real time kinematic global positioning system) was used. Table 1 describes the specification of the radar sensor, and Table 2 shows the RTK-GPS. Figure 9 contains the vehicle environment for data logging. By using the VN 1640 (Vector), in-vehicle

Table 1. Specification of the radar.

Maximum range	150 m
FOV	+/– 10 degrees over 60 m +/– 45 degrees under 60 m
Update rate	50 milliseconds

Table 2. Specification of the RTK-GPS.

Accuracy (Root mean square)	2 cm
Update rate	20 milliseconds

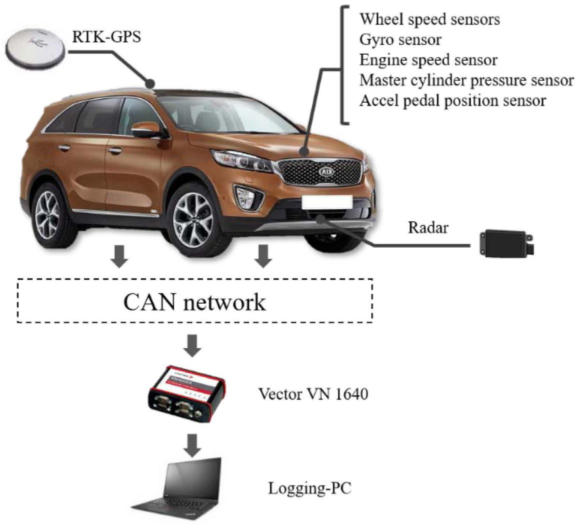


Figure 9. Data logging environment.

information could be acquired through a CAN network. Vehicle speed and acceleration were obtained through wheel sensors and a gyro sensor, and the engine speed was transmitted from the engine control unit signal. The relative speed and distance to the preceding vehicle were recognized through a radar sensor. In the case of the ego-vehicle location on the driving route, an extra process to get the information was needed. It requires a roadway model that can represent the driving route and a proper coordinate system to indicate the location of the vehicle effectively. For this purpose, the B-spline road model (Jo and Sunwoo, 2013; Jo *et al*., 2017) and the curvilinear coordinate system (Wang *et al*., 2002) were utilized.

4. RESULT AND ANALYSIS

The performance of the vehicle speed prediction was evaluated for the six driving tests. In order to evaluate the performance of prediction model, R-squared (R^2) and RMSE were employed. Both R^2 and RMSE were used in this study as an indicator of the model prediction accuracy. R^2 and RMSE were formulated as the following equations:

$$\bar{y} = \frac{1}{N}\Sigma_{t=1}^{N} y_t \qquad (15)$$

$$R^2 = 1 - \frac{\Sigma_{t=1}^{N}(y_t - \hat{y}_t)^2}{\Sigma_{t=1}^{N}(y_t - \hat{y}_t)^2} \qquad (16)$$

$$\text{RMSE} = \sqrt{\frac{\Sigma_{t=1}^{N}(y_t - \hat{y}_t)^2}{N}} \qquad (17)$$

where $\bar{y}$ is the mean of the measured speed, y_t is the measured speed, $\hat{y}_t$ is the predicted speed, and N is the number of elements in output.

The result of the model optimization process is described in 4.1. Comparison results with existing models are shown in 4.2. The proposed vehicle speed model using all inputs is evaluated in 4.3. The influence of the input combinations on prediction is analyzed in 4.4.

4.1. Model Structure Optimization Process

The number of LSTM layers was fixed by 2. The optimal number of hidden states for each layer was found by the HORD algorithm. Table 3 shows the range of the hyperparameters to be optimized. We executed HORD algorithms with three trials within the iteration number of 100. The result is shown in Figure 10 and summarized in Table 4. As shown in Figure 10, the best hyperparameters were observed in Trial 2, iteration 77. The number of best hidden states was 230, 310 for each LSTM layers. Figure

Table 3. Range of the hyperparameters to be optimized.

Hyperparameters	Range
H1	[1, 321]
H2	[1, 321]

*The number of hidden states for 1-LSTM layer is H1, and the number of hidden states for 2-LSTM layer is H2

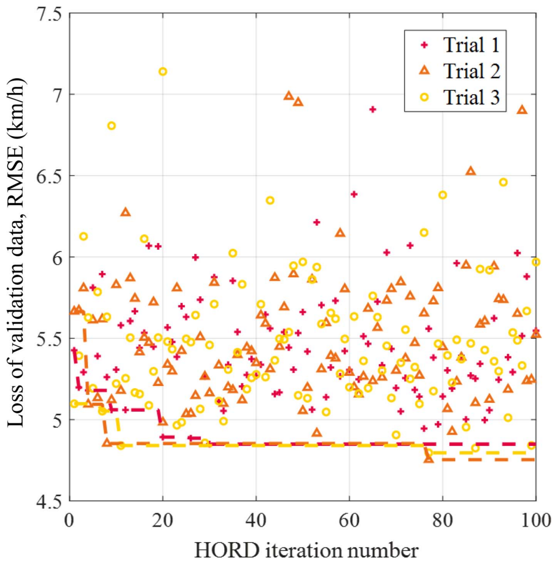


Figure 10. Result of the hyperparameter optimization by HORD algorithm.

Table 4. Summary of the hyperparameter optimization.

Index	Best iteration number	H1	H2	Average RMSE in validation data (km/h)
Trial 1	30	218	125	4.849
Trial 2	77	230	310	4.753
Trial 3	76	270	270	4.796

*The number of hidden states for 1-LSTM layer is H1, and the number of hidden states for 2-LSTM layer is H2

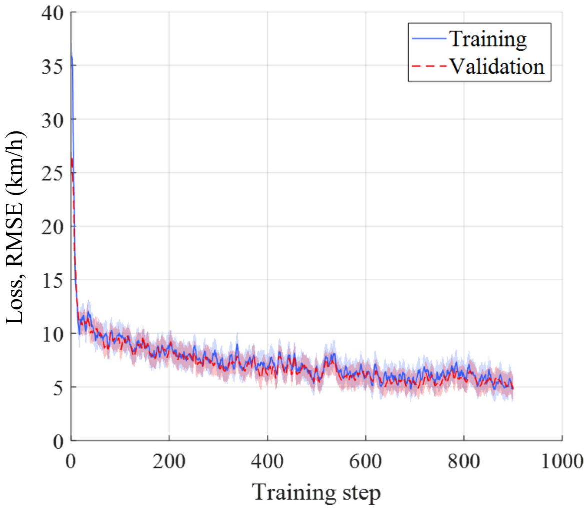


Figure 11. Result of best training at Trial 2, iteration 77.

11 shows the training result when the best hidden state numbers were applied. The average RMSE was 4.995 km/h in training data and 4.753 km/h in validation data. In this way, we optimized the hyperparameters through the HORD algorithm and predicted the speed by using this model.

4.2. Comparison of the Speed Prediction Models

Because accurate prediction is a mandate for predictive powertrain control, the proposed model was compared with CS, CA, ANN. The CS and CA belong to the parametric approach, while the ANN and the proposed model belong to the non-parametric approaches. The CS model assumes that the velocity of the ego-vehicle is constant. The CA model assumes that the ego-vehicle keeps the acceleration constant for the prediction horizon of 15 sec. These two models have been used to compute the time-to-collision and applied to vehicle tracking application when there is no information about other vehicles (Lefèvre *et al*., 2014). The ANN consists of two hidden layers and one output layer, which is the same structure as the proposed model. The optimal hyperparameters of the ANN were obtained through HORD optimization.

The performance of the proposed model was compared with the CA, CS, ANN as shown in Figure 12. The non-parametric approaches show more accurate prediction performance than the parametric approaches regarding R^2 and RMSE as expected. The parametric methods cannot consider various driving environments because their input was only the velocity or acceleration.

On the other hand, the non-parametric methods utilize radar sensor, location information, and internal vehicle sensors. For the non-parametric methods, the proposed model shows more accurate than the ANN. The average RMSE of the proposed model along the prediction horizon was 4.797 km/h compared to that of the ANN at 7.012 km/h. This comes from the vehicle speed of the proposed model is determined by not only the current information but also the continuous sequential driving information from the past. However, the ANN only utilizes the current information. Therefore, we chose the proposed model to predict the vehicle speed in this study.

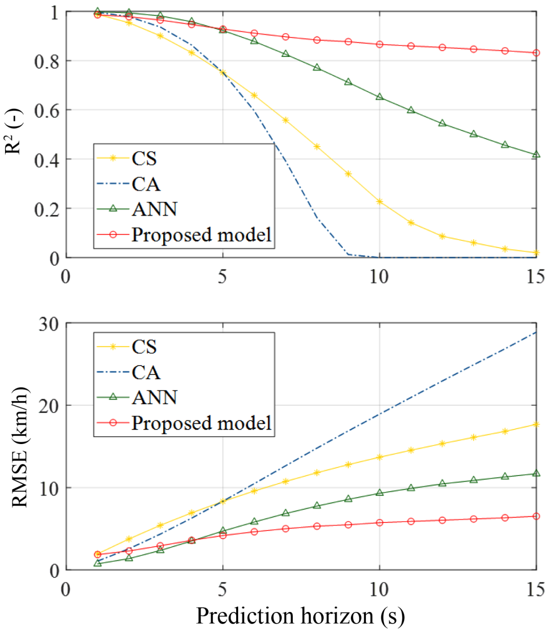


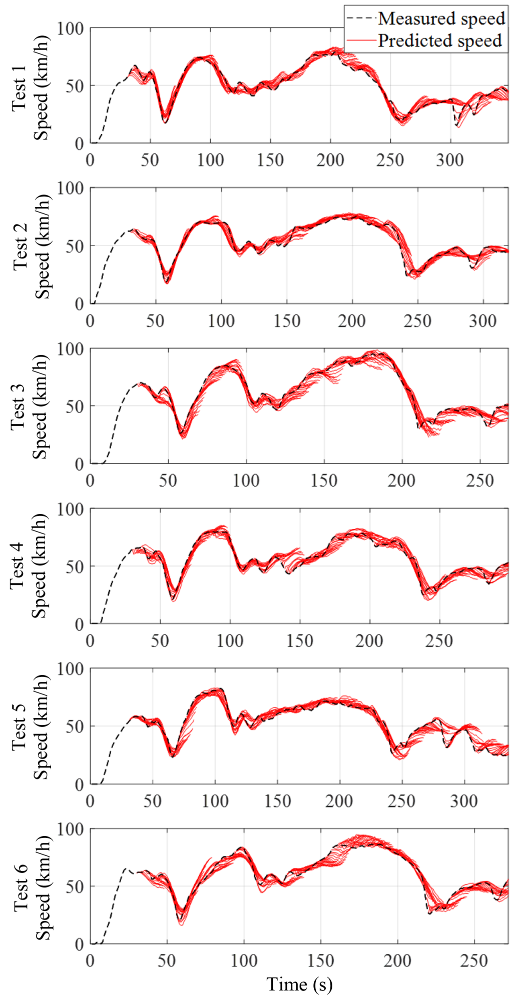
Figure 12. Comparison results with CS, CA, ANN along the prediction horizon.

4.3. Evaluation of the Proposed Model

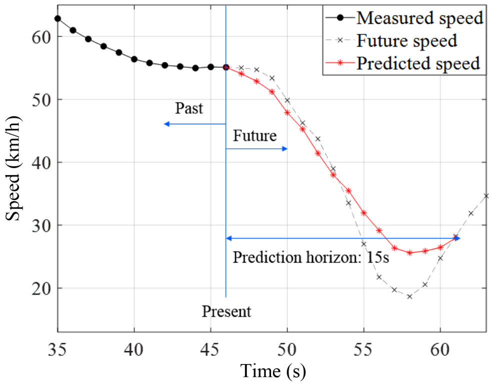
In this section, we evaluate the proposed model in terms of R^2, RMSE. The prediction results of the proposed model for the 6 driving tests are shown in Figure 13. The length of the input states was 30 seconds. Thus, the first prediction takes place after 30 seconds as shown in Figure 13 (a). The black dotted line is the actual speed of the ego-vehicle, and the red lines are the predicted vehicle speed profiles. The red colored prediction profile is generated at present, and it predicts vehicle speed with a 15 second prediction horizon. In order to show the result more clearly, Figure 13 (b) shows the prediction result at 46 seconds of Test 2 in detailed view. For evaluation of the prediction, the R^2 and RMSE were calculated for the 6 driving tests. In addition, the average results of R^2 and RMSE for all the six driving tests were determined as shown in Table 5. The average R^2 in the six driving tests was 0.897, and the average RMSE was 5.023 km/h. The maximum RMSE was 5.820 km/h, and the minimum was 4.030 km/h. The standard deviation of the R^2 was 0.0156 and that of the RMSE of 0.6356 km/h between the result from Test 1 to Test 6.

4.4. Effect of the Input Combinations on the Prediction Performance

As a reminder, G1 is the input group of the internal vehicle information from the on-board sensors, G2 includes the relative speed and distance from the radar sensor, and G3 is



(a)



(b)

Figure 13. (a) Prediction results of the proposed model; (b) Detailed view of the prediction result at the certain moment.

the location of the ego-vehicle from the RTK-GPS. To investigate the influence of these input groups on the prediction results, the driving data was divided into three specific scenarios.

Table 5. R^2 and RMSE in the test set.

Index	R^2 (-)	RMSE (km/h)
Test 1	0.917	4.572
Test 2	0.918	4.030
Test 3	0.894	5.820
Test 4	0.885	4.725
Test 5	0.895	5.334
Test 6	0.875	5.660
Average	0.897	5.023
Standard deviation	0.0156	0.6356

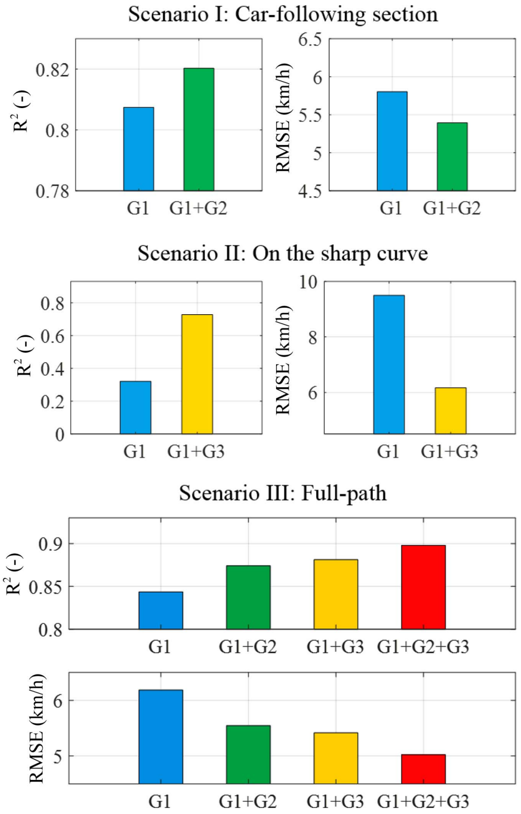


Figure 14. R^2 and RMSE according to input combinations.

Scenario I: Car-following
Scenario II: On the sharp curve
Scenario III: Full path

Scenario I was classified using the radar sensor of the vehicle. This is the section where the relative distance from the front vehicle to the self-vehicle is less than 140 m. Scenario II is a sharp curve. The sharp curve is a corner that has a maximum curvature of 0.02 or more. It is illustrated in Figure 8. Scenario III is the full path of the driving route. Figure 14 shows the results of predictions according to three scenarios. In scenario I, which is affected by the vehicle ahead, the prediction accuracy was improved when G2 was included as inputs. It shows that input from the radar sensor could improve R^2 by 1.6 % and RMSE by 7.0 % in the car-following scenario. For scenario II, the input combination of G1 and G3 provided the best prediction accuracy. It improved R^2 by 55.9 % and reduced the RMSE by 35.0 % compared to only G1 used as input. This result means that the location information of the ego-vehicle significantly increased the performance of the prediction on the curve. Lastly, the input combination of G1, G2, and G3 had the most accurate prediction performance in the scenario III (full path). R^2 was improved by 6.0 % and RMSE reduced by 18.7 % from the only G1 input. Considering all the scenarios, the radar sensor and the information of the location of the ego-vehicle contribute to improvement of the speed prediction accuracy. Thus, we conclude that for application of the predictive powertrain control, the radar sensor and the location of the ego-vehicle information should be concerned as inputs to the speed prediction model as well as the internal vehicle information.

5. CONCLUSION

In this paper, an ego-vehicle speed prediction model using an LSTM based RNN was proposed to improve the prediction performance in various driving conditions. The proposed model uses several model inputs: internal vehicle sensors, a radar sensor, and the location information of the ego-vehicle. The hyperparameters of the proposed model were optimized by the HORD algorithm. More detailed findings are as follows:

The prediction performance of the proposed model was compared to the CS, CA, and ANN. The comparison results showed that the proposed model has the most precise prediction performance.

The proposed model was validated by real-driving data from a vehicle equipped with the radar and RTK-GPS. The driving conditions included car-following, a sharp curve road, and full path.

To analyze the impact of the input combinations, the prediction results of three scenarios were compared. The radar sensor improves the prediction performance in the scenario of car-following, and the location of the ego-vehicle enhances the accuracy of prediction on the sharp curve. The best prediction performance on the full path was obtained by using all of the inputs.

ACKNOWLEDGEMENT–This work was financially supported by the BK21 plus program (22A20130000045) under the Ministry of Education, Republic of Korea, the Industrial Strategy Technology Development Program (No. 10039673, 10060068, 10042633, 10079961, 10080284), the International Collaborative Research and Development Program (N0001992) under the Ministry of Trade, Industry and Energy (MOTIE Korea), and National Research Foundation of Korea (NRF) grant funded by the Korean government (MEST) (No. 2011-0017495).

REFERENCES

Bellemans, T., De Schutter, B. and De Moor, B. (2002). Models for traffic control. *J. A* **43**, **3-4**, 13–22.

Berry, D. S. and Belmont, D. M. (1951). Distribution of vehicle speeds and travel times. *Proc. 2nd Berkeley Symp. Mathematical Statistics and Probability*, Berkeley, California, USA.

Blanch Micó, M. T., Lucas Alba, A., Bellés Rivera, T., Ferruz Gracia, A. M., Melchor Galán, Ó. M., Delgado Pastor, L. C., Ruíz Jiménez, F. and Chóliz Montañés, M. (2017). Car following: Comparing distance-oriented vs. inertia-oriented driving techniques. *Transport Policy*, **67**, 13–22.

Gers, F. A., Schmidhuber, J. and Cummins, F. (2000). Learning to forget: Continual prediction with LSTM. *Neural Computation* **12**, **10**, 2451–2471.

Graves, A., Mohamed, A. and Hinton, G. (2013). Speech recognition with deep recurrent neural networks. *Proc. 38th Int. Conf. Acoustics, Speech, and Signal Processing*, Vancouver, Canada.

Guo, J., Luo, Y. and Li, K. (2017). Adaptive neural-network sliding mode cascade architecture of longitudinal tracking control for unmanned vehicles. *Nonlinear Dynamics* **87**, **4**, 2497–2510.

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

Ilievski, I., Akhtar, T., Feng, J. and Shoemaker, C. A. (2016). Efficient hyperparameter optimization of deep learning algorithms using deterministic RBF surrogates. *arXiv Preprint arXiv,* 1607.08316, 1–8.

Jiang, B. and Fei, Y. (2017). Vehicle speed prediction by two-level data driven models in vehicular networks. *IEEE Trans. Intelligent Transportation Systems* **18**, **7**, 1793–1801.

Jo, K. and Sunwoo, M. (2013). Generation of a precise roadway map for autonomous cars. *IEEE Trans. Intelligent Transportation Systems* **15**, **3**, 925–937.

Jo, K., Lee, M., Kim, C. and Sunwoo, M. (2017). Construction process of a three-dimensional roadway geometry map for autonomous driving. *Proc. Institution of Mechanical Engineers, Part D: J. Automobile Engineering* **231**, **10**, 1414–1434.

Jurecki, R., Poliak, M. and Jaśkiewicz, M. (2017). Young adult drivers: Simulated behaviour in a car-following situation. *Promet - Traffic&Transportation* **29**, **4**, 381–390.

Keulen, T. V., Naus, G. and Jager, B. D. (2009). Predictive cruise control in hybrid electric vehicles. *World Electric Vehicle Journal*, **3**, 494–504.

Lang, D., Schmied, R. and Del Re, L. (2014). Prediction of preceding driver behavior for fuel efficient cooperative adaptive cruise control. *SAE Int. J. Engines* **7**, **1**, 14–20.

Lefèvre, S., Sun, C., Bajcsy, R. and Laugier, C. (2014).

Comparison of parametric and non-parametric approaches for vehicle speed prediction. *Proc. IEEE American Control Conf.*, Portland, Oregon, USA.

Lian, J., Liu, S., Li, L., Liu, X., Zhou, Y., Yang, F. and Yuan, L. (2017). A mixed logical dynamical-model predictive control (MLD-MPC) energy management control strategy for plug-in hybrid electric vehicles (PHEVs). *Energies* **10**, **1**, 1–18.

Lim, H., Su, W. and Mi, C. C. (2017). Distance-based ecological driving scheme using a two-stage hierarchy for long-term optimization and short-term adaptation. *IEEE Trans. Vehicular Technology* **66**, **3**, 1940–1949.

Lipton, Z. C., Berkowitz, J. and Elkan, C. (2015). A critical review of recurrent neural networks for sequence learning. *arXiv Preprint arXiv*, 1506.00019, 1–38.

Liu, R., Xu, S., Park, J., Murphey, Y. L., Kristinsson, J., McGee, R., Kuang, M. and Phillips, T. (2011). Real time vehicle speed predition using gas-kinetic traffic modeling. *Proc. IEEE Symp. Computational Intelligence in Vehicles and Transportation Systems (CIVTS)*, Paris, France.

Ma, X., Tao, Z., Wang, Y., Yu, H. and Wang, Y. (2015). Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transportation Research Part C: Emerging Technologies*, **54**, 187–197.

Michael, S., Jan, P., Jurgen, A., Jurgen, O. and Ye, Z. (2016). Dynamic speed trajectory optimization for achieving real world optimum energy consumption. *Proc. FISITA 2016*, BEXCO, Korea.

Morton, J., Wheeler, T. A. and Kochenderfer, M. J. (2017). Analysis of recurrent neural networks for probabilistic modeling of driver behavior. *IEEE Trans. Intelligent Transportation Systems* **18**, **5**, 1289–1298.

Müller, M., Reif, M., Pandit, M., Staiger, W. and Martin, B. (2004). Vehicle speed prediction for driver assistance systems. *SAE Paper No.* 2004-01-0170.

Ngo, V., Hofman, T., Steinbuch, M. and Serrarens, A. (2011). Predictive gear shift control for a parallel hybrid electric vehicle. *Proc. IEEE Vehicle Power and Propulsion Conf.*, Chicago, Illinois, USA.

Odhams, A. M. C. and Cole, D. J. (2004). Models of driver speed choice in curves. *Proc. 7th Int. Symp. Advanced Vehicle Control (AVEC'04)*, Arnhem, the Netherlands.

Panwai, S. and Dia, H. (2007). Neural agent car-following models. *IEEE Trans. Intelligent Transportation Systems* **8**, **1**, 60–70.

Park, J., Li, D., Murphey, Y. L., Kristinsson, J., McGee, R., Kuang, M. and Phillips, T. (2011). Real time vehicle speed prediction using a neural network traffic model. *Proc. Int. Joint Conf. Neural Networks*, San Jose, California, USA.

Pascanu, R., Mikolov, T. and Bengio, Y. (2012). On the difficulty of training recurrent neural networks. *arXiv Preprint arXiv*, 1211.5063, 1–9.

Rezaei, A. and Burl, J. B. (2015a). Effects of time horizon on model predictive control for hybrid electric vehicles. *IFAC-PapersOnLine* **48**, **15**, 252–256.

Rezaei, A. and Burl, J. B. (2015b). Prediction of vehicle velocity for model predictive control. *IFAC-PapersOnLine* **48**, **15**, 257–262.

Rumelhart, D. E., Hinton, G. E. and Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, **323**, 533–536.

Selvin, S., Vinayakumar, R., Gopalakrishnan, E. A., Menon, V. K. and Soman, K. P. (2017). Stock price prediction using LSTM, RNN and CNN-sliding window model. *Proc. Int. Conf. Advances in Computing, Communications and Informatics (ICACCI)*, Udupi, India.

Serna, C. G. and Ruichek, Y. (2017). Dynamic speed adaptation for path tracking based on curvature information and speed limits. *Sensors* **17**, **6**, 1–22.

Stanger, T. and del Re, L. (2013). A model predictive cooperative adaptive cruise control approach. *Proc. IEEE American Control Conf.*, Washington, USA.

Sun, C., Hu, X., Moura, S. J. and Sun, F. (2015a). Velocity predictors for predictive energy management in hybrid electric vehicles. *IEEE Trans. Control Systems Technology* **23**, **3**, 1197–1204.

Sun, C., Moura, S. J., Hu, X., Hedrick, J. K. and Sun, F. (2015b). Dynamic traffic feedback data enabled energy management in plug-in hybrid electric vehicles. *IEEE Trans. Control Systems Technology* **23**, **3**, 1075–1086.

Tran, V. T., Nguyen, K. H. and Bui, D. H. (2016). A Vietnamese language model based on recurrent neural network. *Proc. 8th Int. Conf. Knowledge and Systems Engineering (KSE)*, Hanoi, Vietnam.

Vlahogianni, E. I., Golias, J. C. and Karlaftis, M. G. (2004). Short-term traffic forecasting: Overview of objectives and methods. *Transport Reviews* **24**, **5**, 533–557.

Vlahogianni, E. I., Karlaftis, M. G. and Golias, J. C. (2014). Short-term traffic forecasting: Where we are and where we're going. *Transportation Research Part C: Emerging Technologies* **43**, **1**, 3–19.

Wang, H., Kearney, J. and Atkinson, K. (2002). Robust and efficient computation of the closest point on a spline curve. *Proc. 5th Int. Conf. Curves Surfaces*, Saint-Malo, France.

Ye, Q., Wong, S. C. and Szeto, W. Y. (2010). Short-term traffic speed forecasting based on data recorded at irregular intervals. *Proc. 13th Int. IEEE Conf. Intelligent Transportation Systems*, Funchal, Portugal.

Zhang, S., Luo, Y., Wang, J., Wang, X. and Li, K. (2017). Predictive energy management strategy for fully electric vehicles based on preceding vehicle movement. *IEEE Trans. Intelligent Transportation Systems* **18**, **11**, 3049–3060.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.