Predictive Intelligent Driver Model using Deep Learning-based Prediction of Surrounding Vehicle Trajectory

Honggyu Lee, Dongyeon Yu, Taeyun Kim, Hyobin Kim and **Sung-Ho Hwang***
Department of Mechanical Engineering, Sungkyunkwan University, Republic of Korea
{ghdrb9138, yunotions2, xodbs2474, illusoriso, hsh0818}@skku.edu

Abstract

As automated driving technology advances, many vehicles are equipped with Advanced Driver Assistance System (ADAS) technologies. Adaptive Cruise Control (ACC) is a representative ADAS technology, and its performance is improving further as various control models are developed. However, the simple following model shows poor performance in actual road due to unexpected lane changes of nearby vehicles. In this paper, we propose a safer, more economical, and more comfortable car-following model, called the Prediction Intelligent Driver Model (P-IDM), using Deep-learning to predict the trajectory of nearby vehicles for the next 5 seconds. The basic following model is the Intelligent Driver Model (IDM). In the case of deep learning to predict the trajectory of surrounding vehicles, the HighD Dataset and the Long-Short Term Memory Encoder-Decoder (LSTM seq2seq) model are used to train the data. As a result of the experiments using simulation, the proposed following model could have a better cruise control performance than the standard IDM model in economy and safety.

1 Introduction

As autonomous driving technology develops daily, level 4 is becoming a reality beyond level 3 [J. Shuttleworth, 2019]. As the level of autonomous driving increases, various functions are being added. Since the past, Adaptive Cruise Control (ACC) has been evaluated as a key technology in autonomous driving technology along with Advanced Driver Assistance Systems (ADAS). Confidence in ACC is improving as recognition technology for surrounding vehicles improves with better sensors. ACC has various advantages, such as driver convenience, prevention of traffic jams, the effect of increasing ride comfort, driving fuel efficiency, and safety against collisions [Wang et al., 2004; Ploeg et al., 2011; Milanés et al., 2013]. Although, in the past, ACC could only be used in certain driving situations, such as highways, the efficiency of cruise control is becoming more important today as more and more vehicles are equipped with it.

Various following models have been proposed, starting with traditional following models, such as the Optical Velocity (OV), Generalized Force (GF) model, Full Velocity Difference (FVD) model, and Adaptive Control models using a connected automated vehicle (CAV) [Ahmed et al., 2021; Zhu et al., 2018; Olstam et al., 2004]. Among these many models, we used the Intelligent Driver Model (IDM), a reasonable basis for ACC system development [Treiber et al., 2000]. This model exhibits controllable stability characteristics and implements an intelligent braking strategy by modifying the smooth transition between acceleration and deceleration. However, this model also has several drawbacks. They do not investigate how nearby vehicles affect the driving status of ego vehicles, nor do they consider the effects of driving characteristics such as time delays occurring in actual vehicle hardware.

Modified IDM models are continuously developed to overcome these problems. Human Driver Model (HDM)-IDM combination was proposed to allow accident-free smooth driving in complex traffic situations [Treiber et al., 2006]. And The model, Modified IDM, which considers the capability of actual vehicles also proposed. [Derbel et al., 2013] Several IDM models are developed under the assumption of connected vehicles, called V2V. In the situation of On-ramping merge, a Cooperative IDM model applied concerning the movement of surrounding vehicles from on-ramping point [Zhou et al., 2016] and multi-front/rear vehicles in the same lane are considered to apply in the IDM have also been proposed [Zong et al., 2021].

As previous studies have shown, cruise control is eventually heavily influenced by surrounding vehicles, especially vehicles in front of the ego vehicle, and various methods have been devised to minimize problems arising from surrounding effects. One of the typical methods is to predict the trajectory of nearby vehicles. There are various methods of predicting the route of nearby vehicles, and in recent years, research using Deep-Learning (or Machine-Learning) has become the trend of trajectory prediction method [Mozaffari et al., 2020]. Several datasets are built for this, and various deep learning models are developed. It is developed in various ways, including Gaussian Mixture Models, [Wiest et al., 2012] Hidden Markov Model [Deo et al., 2018] and Recurrent Neural Networks (Encoder-Decoder) [Yu et al., 2021; Deo and

Trivedi, 2018; Han et al., 2019], as well as the Convolutional Neural Network [Nikhil and Morris, 2018].

The purpose of this study is to predict the trajectory of nearby vehicles using the basic LSTM Encoder-Decoder (seq2seq) model and to show that the tracking ability of ego vehicles is improved using the predicted trajectory. Other studies also used the predicted trajectory or lane change tendency for Cruise-Control [Lee et al., 2017; Tang et al., 2008; Yuan et al., 2018]. However, we propose better performance algorithm with more intuitive and more straightforward equations than the previous ones and analyze it by applying them to vehicle dynamics simulation.

This paper consists of the following sections. Section 2 describes the datasets preprocessing and trajectory prediction deep-learning model and shows the results of the predicted trajectory of surrounding vehicles in the simulation. Section 3 introduces P-IDM, which is used by attaching the IDM and the trajectory prediction model. Section 4 shows the results of the experiment through simulation. Finally, in Section 5, this study is summarized as a conclusion.

2 Trajectory Prediction

2.1 HighD Dataset

In many related studies, there are various datasets [Colyar and Halkias, 2006; Colyar and Halkias, 2007] to learn and verify the surrounding vehicle trajectory prediction model. In this paper, we used Open-source Dataset called the HighD Dataset, which is a high-precision, large-scale traffic dataset [Krajewski et al., 2018]. The dataset includes 110,000 post-processing trajectories, including cars and trucks (a large-scale naturalistic vehicle) extracted through drone video recording on German highways. This dataset contains data including vehicle size, type, driving direction, position, velocity, acceleration, the number of lane changes, and ID, automatically extracted from all vehicles recorded by drones using computer vision algorithms like Figure 1.

This dataset has the advantage of including vehicle information around the target vehicle and information describing various driving situations such as lane change and having high-quality raw data with an error of less than 10cm. We downsampled 25-fps data to 5-fps to reduce unnecessary information and divided training, validation and testing sets with a ratio of 7:1:2.

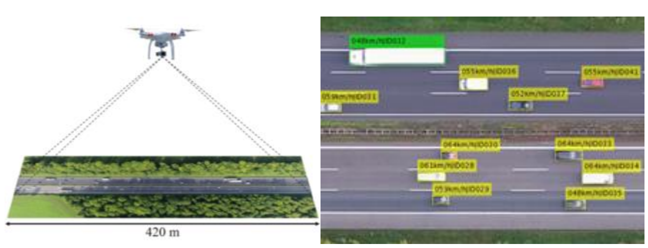

Figure 1: HighD Dataset: highway traffic dataset [Krajewski et al., 2018]

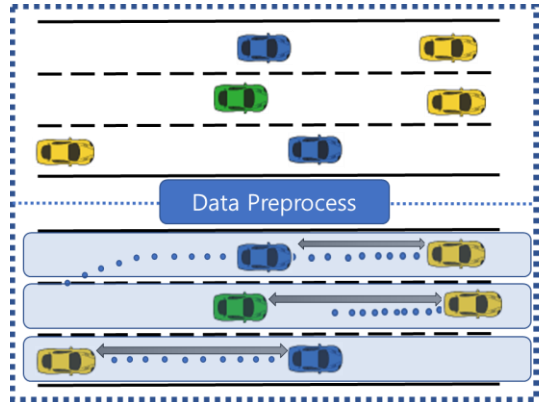

Figure 2: HighD Dataset Preprocess Example

2.2 Surrounding Vehicles Data Processing

We preprocessed the data to be used in the deep learning model using the HighD dataset. The x,y coordinates of the HighD dataset are used as the reference (x: heading direction / y: perpendicular to heading). In Figure 2, when a green vehicle is a target vehicle to predict, history data (points) of a nearby vehicle and relative relationship information are stored. In the previous studies, the time it usually takes to change lanes while driving is about 5 seconds [Han et al., 2019]. Therefore, we set the time of the predicted path to 5 seconds and used the data from the last 3 seconds at 5 Hz for this purpose.

2.3 LSTM Encoder-Decoder (Seq2Seq) Model

The Long-Short Term Memory (LSTM) Model, Figure 3 (b), is a type of Recurrent Natural Networks (RNN), Figure 3 (a), and is a model developed by adding a 'Gate' to overcome the disadvantages of RNN. Here, the disadvantage of RNN is that there is a limit to learning a long-term dependence relationship (; Vanishing Gradient Problem), which is temporally far from time-series data. The LSTM uses cell-state (; called 'memory-cell'), which goes through four processes (forget/input/update/output) to accumulate past data by properly combining past information and new input information.

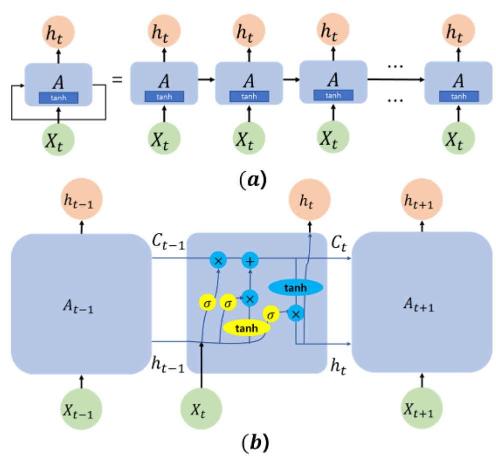

Figure 3: (a) RNN (b) LSTM standard architecture [Yellow : Neural Network Layer / Blue : Pointwise Operation]

- Forget gate: Where to save any of the current information in Cell State (sigmoid & tanh)
-
$$f_t = \sigma(W_{xh_f}x_t + W_{hh_f}h_{t-1} + b_{h\_f}) \quad (1)$$

- Input gate: Where to decide what information to throw away in the previous state through the sigmoid layer
-
$$i_t = \sigma(W_{xh_i}x_t + W_{hh_i}h_{t-1} + b_{h\_i}) \quad (2)$$

- Update Cell-state: Where information that has passed through Forget and Input gate is updated.
-
$$g_t = tanh(W_{xh_g}x_t + W_{hh_g}h_{t-1} + b_{h\_g}) \quad (3)$$

- Output gate: Where the output values are made from the long-term(C_{t-1}) and short-term(h_{t-1}) states
-
$$o_t = \sigma(W_{xh_o}x_t + W_{hh_o}h_{t-1} + b_{h\_o}) \quad (4)$$

where σ is an activation function, W_x and W_h represent weight matrix for input and hidden state. x_t is input vector at time step t and b is bias vector of each gate.

The LSTM Encoder-Decoder (Seq2Seq) model is originally developed as a machine translation field [Sutskever *et al.*, 2014]. However, in this paper, the trajectory prediction model was implemented using the model as mentioned above, illustrated in Figure 4 [Yu et al., 2021; Deo and Trivedi, 2018]. A detailed description is as follows:

① Encoder-Decoder (Sequence to Sequence)

The encoder-decoder model is created along with the basic structure. The number of n-Layer is 2, and the teaching force, scaler, and drop-out are used to overcome the overfitting of model training.

② Input-Output

$$X_{input} = [x^{t-14}, x^{t-13}, \dots, x^{t-1}, x^t] \quad (5)$$

$$x^t = [x^t_{i,c}, x^t_{i,l}, x^t_{i,r}] \quad (6)$$
$$[x_i: x^t_m, y^t_m, v^t_{x,m}, v^t_{x,m}, x^t_f, y^t_f, d^t_{mf}, \Delta v^t_{mf}, x^t_r, y^t_r, d^t_{mr}, \Delta v^t_{mr}]$$

$$Y_{output} = [y^t, y^{t+1}, \dots, y^{t+24}] \quad (7)$$

$$y^t = [x^t_p, y^t_p] \quad (8)$$

Input data used in the proposed model contains information on the location coordinates of the target vehicle and the relationship, such as relative distance (d) and relative velocity (Δv), between left and right (c: center, l: left, r: right) and front and rear vehicles (m: middle, f: front, r: rear). When preprocessed data is input for 3 seconds at a 5Hz cycle (X_{input}: 15 data), the predicted coordinates x, y for the next 5 seconds (Y_{output}: 25 data) are obtained as output.

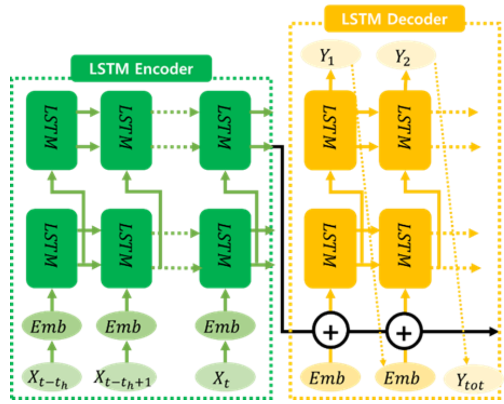

Figure 4: LSTM Encoder-Decoder architecture

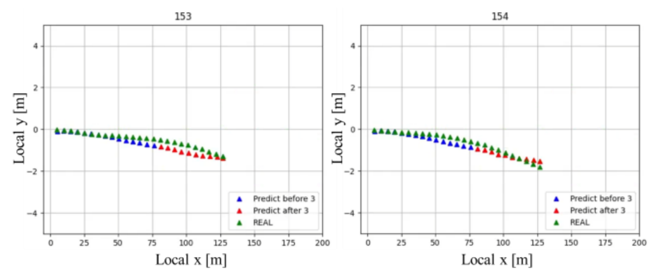

Figure 5: Example of continuous prediction of vehicle routes

③ Loss function

$$RMSE_{loss} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\sum_{t=1}^{t_f}\{(\hat{x}-x)^2 + 2\cdot(\hat{y}-y)^2\}} \quad (9)$$

To obtain the learning effect of the deep learning model, loss function is well-selected and used. Since we are a problem related to distance, we use the modified Root Mean Square Error ($RMSE_{loss}$) method shown in (9). Here, $\hat{x}$, $\hat{y}$ are predicted position coordinates, and x, y are true position coordinates, and to further weight the lateral path prediction, the error between y and $\hat{y}$ in (9) is multiplied by 2. In this way, an error is obtained through the above calculation every time-step, and backpropagation is performed.

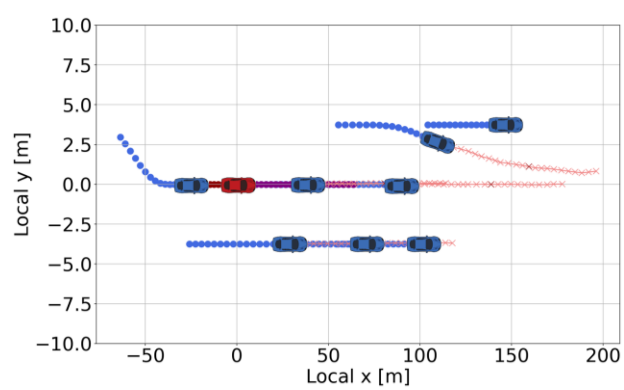

Figure 6: Example of trajectory prediction of vehicles close to the ego vehicle with a simulation scenario

2.4 Surrounding Vehicle Trajectory Prediction Method & Result

We first conducted a dataset-based evaluation to assess the proposed model. In Figure 5, which shows the prediction results in real-time, the path is predicted like blue marks (a prediction path for the future 3 seconds) and red marks (a prediction path for the future 3 to 5 seconds). Next, we verified the proposed model using the IPG 10.1 CarMaker virtual test-driving environment in other situations. This software is a vehicle dynamic simulator widely used to evaluate vehicle driving performance by implementing the actual road environment and vehicle model. We created several scenarios for path predicting model evaluation in CarMaker, and Figure 6 shows one of the results of predicting the target vehicle's route (Red color & x shape). In order to know more detailed accuracy information of the learning model, the prediction errors were compared and measured in the longitudinal and transverse directions. The results are shown in Figure 7. The orange line is a dataset-based evaluation, and the blue one is a simulation-based evaluation, which showed relatively inaccuracy when applying the model to various scenarios. In addition, both models were found to have increased errors over time.

The important point of this trajectory prediction method is to apply the prediction result for one target vehicle to all surrounding vehicles and convert the predicted trajectory of the surrounding vehicles into the ego vehicle's local coordinate. This is because, when the ego vehicle drives, the path trajectory of the nearby vehicles affects. Therefore, for each step, all surrounding vehicles are used as target vehicles to predict the trajectory, and the predicted trajectories are converted based on the ego vehicle to obtain results.

3 Predictive Intelligent Driver Model

In this section, we analyze the Intelligent Driver Model (IDM), the representative model of Adaptive Cruise Control (ACC), and develop the Predictive Intelligent Driver Model (P-IDM), which can overcome the shortcomings of the standard model by using prediction of the surrounding vehicle trajectory.

3.1 Intelligent Driver Model

The IDM is a type of optical velocity (OV) model, a time-continuous car-following model that considers the speed of the ego vehicle, the relative distance, and the relative speed of the vehicle in front of it. The acceleration by the IDM is calculated as follows:

$$\dot{x_\alpha} = \frac{dx_\alpha}{dt} = v_\alpha \tag{10}$$

$$\dot{v_\alpha} = \frac{dv_\alpha}{dt} = a\left(1 - \left(\frac{v_\alpha}{v_0}\right)^{\delta} - \left(\frac{s^*(v_\alpha,\Delta v_\alpha)}{s_\alpha}\right)^2\right) \tag{11}$$

$$s^*(v,\Delta v) = s_0 + v_\alpha T + \frac{v_\alpha \Delta v_\alpha}{2\sqrt{ab}} \tag{12}$$

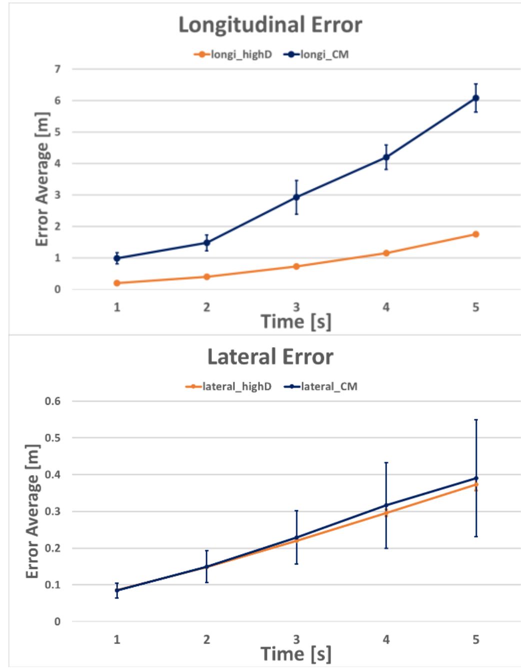

Figure 7: Longitudinal & lateral prediction RMSE in meters (Example Scenario in HighD Dataset-Orange line & CarMaker Software-Blue line)

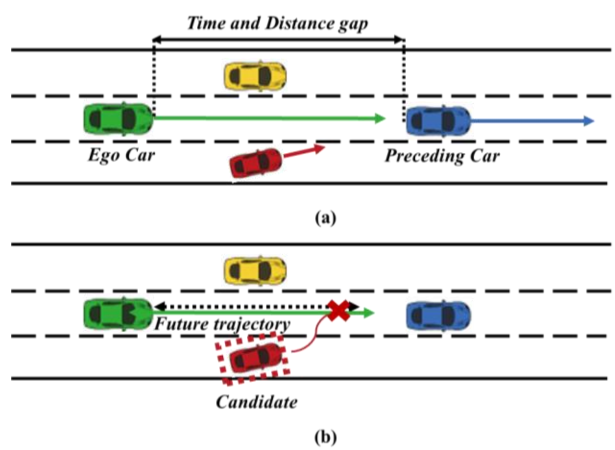

Figure 8: Example Driving Situation. (Green : Ego vehicle, Blue : front vehicle, Red : Lane Change vehicle & (a) Relative Relationship with Preceding Car / (b) Candidate based on Trajectory prediction)

- v_0 & $v_\alpha(\Delta v_\alpha)$: Desired velocity in free traffic & Current Velocity of Ego vehicle (Relative velocity of the vehicle in front)
- s_0 & s_α : Minimum desired net distance & Relative distance of the vehicle in front (bumper-to-bumper gap)
- a & b : Maximum Acceleration & Comfortable Braking Deceleration
- T & δ : Minimum possible time to the vehicle in front & Acceleration exponent

Variable	Realistic Bound [Unit]	Value
v_0	60 ~ 140 $[km/h]$	120
δ	-*	4
T	1 ~ 3 $[s]$	1.5
a	0.5 ~ 2 $[m/s^2]$	0.73
b	0.5 ~ 2 $[m/s^2]$	1.67
s_0	1 ~ 5 $[m]$	2

Table 1: Summary of the IDM parameters [Malinauskas, 2014] {* ; δ characterize how the acceleration/ deceleration work with ego vehicle's velocity (δ =1 : linear work / $\delta = \infty$: constant work)}

Table 1 presents the IDM parameters that was used in this paper. The characteristic of the IDM is as follows:

- **Accident-free**: it is constructed by dependence on the relative velocity of the leading vehicle.

- **Strategies for Traffic Situations**: the maximum required speed is applied to facilitate application even in free traffic situations, where $[s_\alpha \approx \infty]$ makes $[\dot{v_{free}} = a\,(1 - \left(\frac{v_\alpha}{v_0}\right)^{\delta}]$. And in traffic situations, safety time gap, maximum acceleration, and comfortable braking deceleration are applied to show appropriate follow-up capabilities for each situation.

- **Usefulness of model parameters**: All model parameters are interrelated and have a reasonable interpretation. In addition, it is empirically measurable and has an order of expected sizes, so that the parameters and stability characteristics of the model can be easily (and separately) corrected for empirical data.

- **Simulation Analysis**: easy and fast numerical simulation.

However, since the IDM is an equation based on the assumption that it is a one-lane situation, Lane-Change (or Cut-in maneuvers) of nearby vehicles can cause a dangerous situation. In high-speed driving situations, automated vehicles are difficult to respond to the movements of nearby vehicles. When an unexpected lane change comes in urgently from a nearby vehicle, the parameters related to the acceleration equation change discontinuously and rapidly. This penalizes the ride comfort and vehicle performance with more deceleration than necessary. [= The IDM brakes stronger than b, comfortable deceleration, in emergencies] There are more diverse situations and driving tendencies in the actual road situation. It may lead to strong braking maneuvers of the IDM, which would not be acceptable (nor possible) in a real-world ACC system. Therefore, solving these problems is what cruise control have to overcome.

3.2 Predictive-IDM

As mentioned in the previous section, the final acceleration value of the IDM is calculated using the preceding vehicle's relative speed and distance. It means that the relative information with the leading vehicle is important when calculating the final value. As shown in Figure 8 (b), when an adjacent vehicle changes lane, the relative relationship value changes discontinuously (or rapidly), causing following control problems.

This problem is intended to be minimized through trajectory prediction of adjacent vehicles. We changed the relative distance and speed values based on the predicted path as shown in the following equations (13) ~ (18). To introduce it in order based on Figure 9, in Equation (13), we get a gain called '*p*', which is an expression that the closer to 0, the more weighted it is to the actual value, not the predicted value. From the molecular value, if the vehicle is predicted to move to the next lane, the difference between the predicted y ($y_p^{t_{second}}$) and the current y ($y^{current}$) becomes significant and '*p*' increases. It means that the ratio of using the predicted value increases. On the other way, if the vehicle is predicted not to move, the '*p*' decreases, and the predicted value is not used. In addition, as seen in the previous section, the time value (t_{second}) was added to the denominator value because the prediction error was significant in proportion to time. It was also used as a criterion for confirming how much influence predicting time in advance in P-IDM has on follow-up control. The obtained value, '*p*', was applied to (14) and (16) to make the P-IDM parameter ($\hat{s}_\alpha, \hat{v}_\alpha, \Delta\hat{v}_\alpha, \hat{s}^*(\hat{v}_\alpha, \Delta\hat{v}_\alpha)$) with the front car's acceleration (a_α), the predicted relative distance ($s_\alpha^{t_{second}}$) and the predicted relative speed ($\Delta v_\alpha^{t_{second}}$). To illustrate the situation in Figure 10 as an example, the value of '*p*' will rise while the probability of lane change of adjacent vehicles is obvious, and thus the P-IDM parameter values ($\hat{s}_\alpha, \hat{v}_\alpha$) used by the rising '*p*' will be lowered. The P-IDM target acceleration ($\dot{v}_{\alpha[P-IDM]}$) obtained by applying the predicted values is as follows:

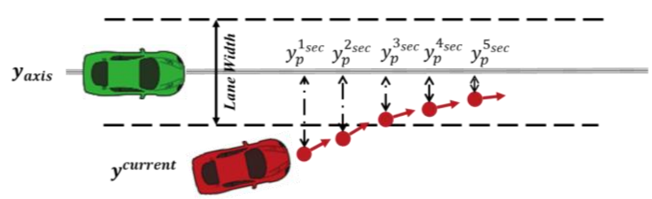

Figure 9: Example of the predicted y($y_p^{t_{second}}$) by time value(t_{second})

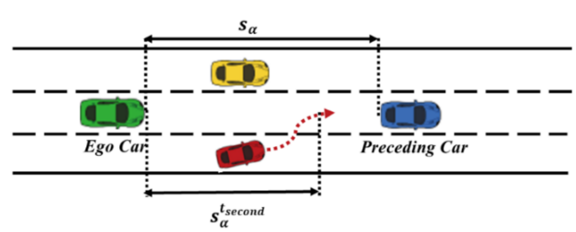

Figure 10: Examples of relative distance (s_α) & predicted relative distance ($s_\alpha^{t_{second}}$)

$$\mathbf{0} \le \boldsymbol{p} = \left(\frac{y_p^{t_{second}} - y^{current}}{\frac{LaneWidth}{2} + \frac{t_{second}}{60}}\right)^2 \le \mathbf{1} \tag{13}$$

$$\hat{\boldsymbol{s}}_{\boldsymbol{\alpha}} = (\mathbf{1} - \boldsymbol{p}) \cdot \boldsymbol{s}_{\boldsymbol{\alpha}} + \boldsymbol{p} \cdot \boldsymbol{s}_{\boldsymbol{\alpha}}^{t_{second}} \tag{14}$$

$$\hat{\boldsymbol{v}}_{\boldsymbol{\alpha}} = \boldsymbol{v}_{\boldsymbol{\alpha}} + \boldsymbol{t}_{\boldsymbol{second}} \cdot \boldsymbol{a}_{\boldsymbol{\alpha}} \tag{15}$$

$$\Delta\hat{\boldsymbol{v}}_{\boldsymbol{\alpha}} = (\mathbf{1} - \boldsymbol{p}) \cdot \Delta\boldsymbol{v}_{\boldsymbol{\alpha}} + \boldsymbol{p} \cdot \Delta\boldsymbol{v}_{\boldsymbol{\alpha}}^{t_{second}} \tag{16}$$

$$\hat{\boldsymbol{s}}^*(\hat{\boldsymbol{v}}_{\boldsymbol{\alpha}}, \Delta\hat{\boldsymbol{v}}_{\boldsymbol{\alpha}}) = \boldsymbol{s}_{\mathbf{0}} + \hat{\boldsymbol{v}}_{\boldsymbol{\alpha}}\boldsymbol{T} + \frac{\hat{v}_\alpha \Delta\hat{v}_\alpha}{2\sqrt{ab}} \tag{17}$$

$$\dot{\boldsymbol{v}}_{\boldsymbol{\alpha}[\boldsymbol{P-IDM}]} = \boldsymbol{a}\left(\mathbf{1} - \left(\frac{\hat{v}_\alpha}{v_0}\right)^{\delta} - \left(\frac{\hat{s}^*(\hat{v}_\alpha, \Delta\hat{v}_\alpha)}{\hat{s}_\alpha}\right)^2\right) \tag{18}$$

4 Experiment Result

4.1 Simulation Environment

We created two scenarios, Cut-in and Cut-out situation, in CarMaker with several vehicles nearby to verify the model. In the case of the experimental road, a three-lane highway was used among the example road models like Figure 11. Moreover, to indicate the realistic limitations of the front object recognition distance, the maximum front vehicle recognition distance was set to 150m or less.

For trajectory prediction, the data about the surrounding vehicle is applied to the trained model in real-time. In the case of the following control (IDM / P-IDM), the experiment was conducted based on the parameters in Table 1, and the time value range ($\boldsymbol{t}_{\boldsymbol{second}}$) was set to a range of 1 to 5 seconds. For the same experimental environment, the driving of nearby vehicles was also made under the same conditions.

4.2 Simulation Results

In ACC, increasing ride comfort and fuel efficiency are commonly required performances, but some functions are required differently in Cut-in and Cut-out situations. In the Cut-in situation, it is necessary to predict that a car in the next lane will enter in front of an automated vehicle and secure a safe distance in advance if it enters. (= *need a longer relative distance*). In the Cut-out situation, it is necessary to know in advance that the vehicle in front will change lanes, and to relieve traffic congestion by quickly approaching the vehicle in front. (= *need a shorter relative distance*). To analyze the ride comfort, we considered the pitch acceleration value from CarMaker. Also, the remaining battery capacity of electric vehicles provided by CarMaker was used for efficiency analysis in each scenario.

In the case of P-IDM, the experiment was conducted for each second, and the results were compared with the IDM. The experimental results for each scenario are shown in Figure 12, 13, 14, and 15, and the analysis results are shown in Table 2. When analyzing each situation in detail, the P-IDM certainly decelerated faster than the existing IDM in the Cut-in scenario, widening the distance from the incoming car in advance. Energy efficiency improved by 1.52 %, 5.21 %, 6.62 %, 3.78 %, and 7.44 % in the order of time, and pitch acceleration, an indicator of ride comfort, showed a slight improvement. In the case of Cut-out, it showed that the relative distance narrowed rapidly through faster acceleration. Similar to the Cut-in situation, energy efficiency improved by 1.23 %, 2.31 %, 2.84 %, 3.38 % and 1.84 %, and pitch acceleration also improved.

5 Conclusion

This paper proposes the Predictive Intelligent Driver Model (P-IDM) that uses the LSTM seq2seq deep-learning model to predict the path of surrounding vehicles and controls them to follow the leading vehicles based on the previously obtained prediction information. The proposed following model is designed to have effective (and efficient) acceleration/deceleration control in situations where surrounding vehicles make lane changes in multi-lane situations, the limitations that the existing following model have.

The HighD dataset, an open dataset of naturalistic vehicle trajectories, was used for trajectories prediction learning of nearby vehicles, and the IDM was used as a basic model for the cruise control model.

For model verification, the experiment was conducted by dividing the virtual environment into two scenarios: Cut-in & Cut-out. As a result of the experiment, it was confirmed that the P-IDM responds faster to surrounding vehicles through real-time trajectory prediction than the standard IDM model. This model has been proven to have better performance, such as safety, ride comfort, economical driving. However, as the results show, they sometimes show different trends, and a faster prediction does not guarantee better results.

In the future, we plan to supplement path prediction by applying more advanced path prediction deep-learning models and various datasets. This will not only increase the reliability of the prediction results, but also can be used universally in more diverse situations. In addition, in the follow-up control strategy, a study will be conducted to create an advanced P-IDM through the application of various considerations such as the predicted speed and predicted location of autonomous vehicles, and the application of parameters considering multiple vehicles.

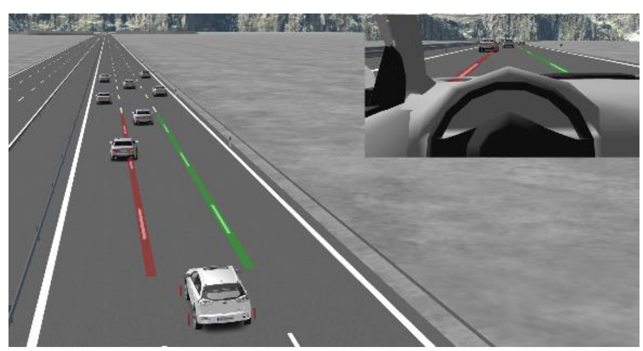

Figure 11: CarMaker Example Road with 8-nearby vehicles

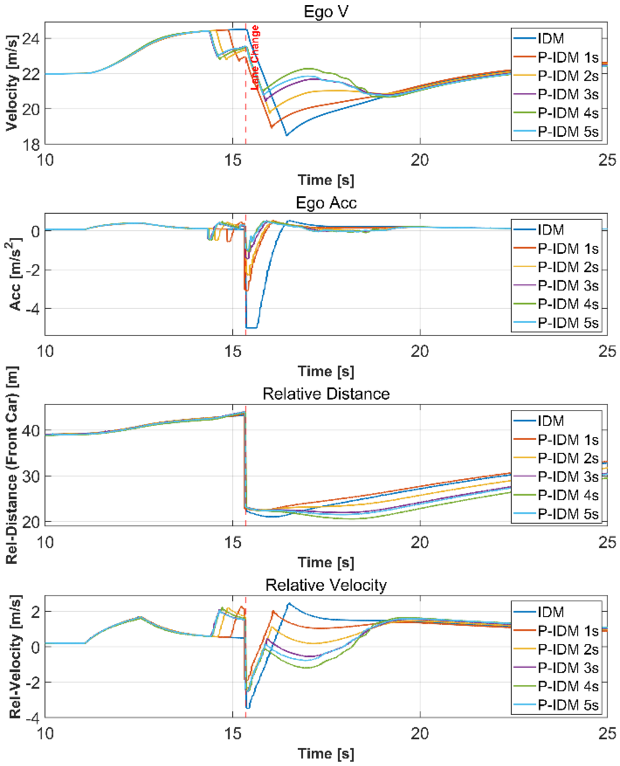

Figure 12: Cut-in Scenario : Used data information

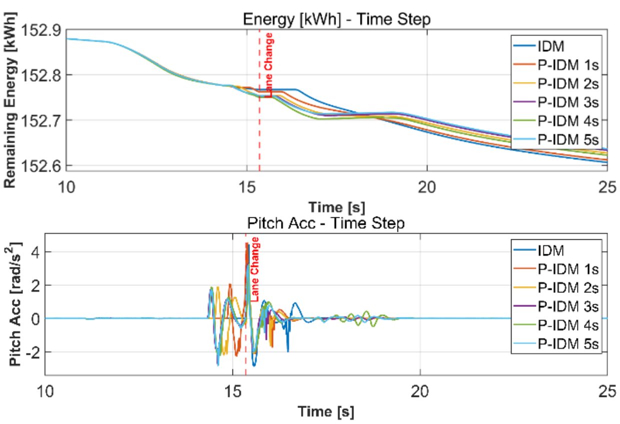

Figure 13: Cut-in Scenario : Result of simulation output

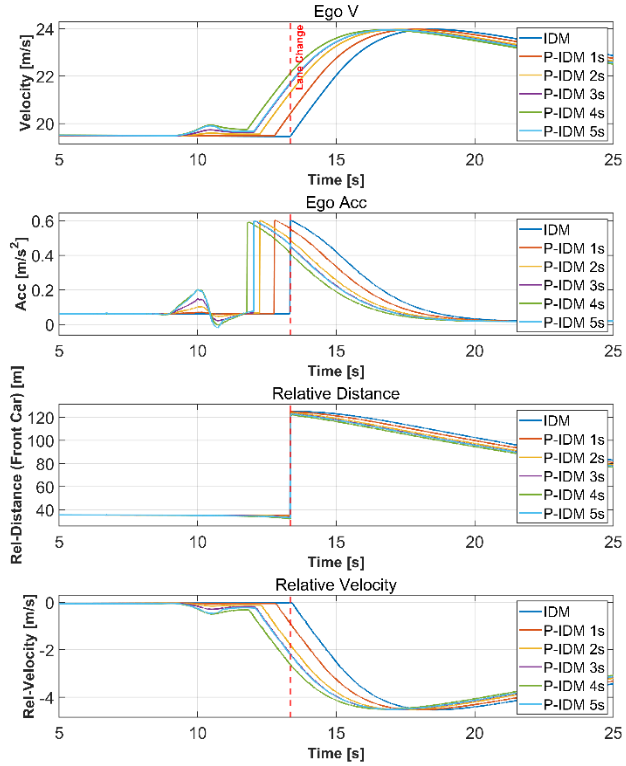

Figure 14: Cut-out Scenario : Used data information

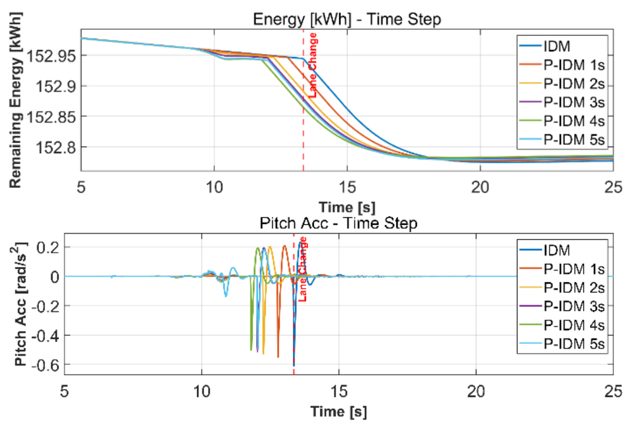

Figure 15: Cut-out Scenario : Result of simulation output

Cut-in Scenario	IDM	P-IDM [1s]	P-IDM [2s]	P-IDM [3s]	P-IDM [4s]	P-IDM [5s]
Max. Deceleration [m/s^2]	-5.02	-3.11	-2.27	-1.44	-1.06	**-0.98**
Avg. Rel Dist after L.C [m]	30.37	30.78	29.12	27.77	**26.57**	27.52
Used Energy [kWh]	0.385	0.379	0.365	0.359	0.371	**0.356**
Pitch Acc Min/Max [rad/s^2]	-2.88/4.44	-2.26/4.55	**-2.18/2.89**	-2.81/3.27	-2.24/3.28	-2.83/3.18
Cut-out Scenario	**IDM**	**P-IDM [1s]**	**P-IDM [2s]**	**P-IDM [3s]**	**P-IDM [4s]**	**P-IDM [5s]**
Max/Avg. Acceleration [m/s^2]	0.596/0.085	0.602/0.084	0.599/0.083	0.598/0.083	**0.591/0.082**	0.597/0.083
Avg. Rel Dist after L.C [m]	93.85	93.43	91.05	89.92	**88.57**	89.71
Used Energy [kWh]	0.204	0.202	0.199	0.198	**0.196**	0.2`
Pitch Acc Min/Max [rad/s^2]	-0.61/0.23	-0.55/0.21	-0.53/0.2	-0.52/0.19	-0.5/0.19	**-0.49/0.19**

Table 2: Result Analysis of Cut-in/Cut-out Scenario

Acknowledgments

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2021-2018-0-01426) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation), and by「the Competency Development Program for Industry Specialists」 of Korean Ministry of Trade, Industry and Energy (MOTIE), operated by Korea Institute for Advancement of Technology (KIAT). (No. P0017120, HRD program for Foster R&D specialist of parts for eco-friendly vehicle(xEV))

References

[J. Shuttleworth, 2019] J. Shuttleworth. SAE Standards News: J3016 automated-driving graphic update. *SAE International*, 2019.

[Wang *et al.*, 2004] J. Wang and R. Rajamani, The impact of adaptive cruise control systems on highway safety and traffic flow, *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 218, no. 2, pp. 111-130, 2004.

[Ploeg *et al.*, 2011] J. Ploeg, B. T. Scheepers, E. Van Nunen, N. Van de Wouw, and H. Nijmeijer, Design and experimental evaluation of cooperative adaptive cruise control, in *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2011: IEEE, pp. 260-265.

[Milanés *et al.*, 2013] V. Milanés, S. E. Shladover, J. Spring, C. Nowakowski, H. Kawazoe, and M. Nakamura, Cooperative adaptive cruise control in real traffic situations, *IEEE Transactions on intelligent transportation systems*, vol. 15, no. 1, pp. 296-305, 2013.

[Olstam *et al.*, 2004] J. J. Olstam and A. Tapani, *Comparison of Car-following models*. Swedish National Road and Transport Research Institute Linköping, 2004.

[Ahmed *et al.*, 2021] H. U. Ahmed, Y. Huang, and P. Lu, "A Review of Car-Following Models and Modeling Tools for Human and Autonomous-Ready Driving Behaviors in Micro-Simulation," *Smart Cities*, vol. 4, no. 1, pp. 314-335, 2021.

[Zhu *et al.*, 2018] M. Zhu, X. Wang, and A. Tarko, Modeling car-following behavior on urban expressways in Shanghai: A naturalistic driving study, *Transportation research part C: emerging technologies*, vol. 93, pp. 425-445, 2018.

[Treiber *et al.*, 2000] M. Treiber, A. Hennecke, and D. Helbing, Congested traffic states in empirical observations and microscopic simulations, *Physical review E*, vol. 62, no. 2, p. 1805, 2000.

[Treiber *et al.*, 2006] M. Treiber, A. Kesting, and D. Helbing, Delays, inaccuracies and anticipation in microscopic traffic models, *Physica A: Statistical Mechanics and its Applications*, vol. 360, no. 1, pp. 71-88, 2006.

[Derbel *et al.*, 2013] O. Derbel, T. Peter, H. Zebiri, B. Mourllion, and M. Basset, Modified intelligent driver model for driver safety and traffic stability improvement, *IFAC Proceedings Volumes*, vol. 46, no. 21, pp. 744-749, 2013.

[Zhou *et al.*, 2016] M. Zhou, X. Qu, and S. Jin, On the impact of cooperative autonomous vehicles in improving freeway merging: a modified intelligent driver model-based approach, *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 6, pp. 1422-1428, 2016.

[Zong *et al.*, 2021] F. Zong, M. Wang, M. Tang, X. Li, and M. Zeng, An Improved Intelligent Driver Model Considering the Information of Multiple Front and Rear Vehicles, *IEEE Access*, vol. 9, pp. 66241-66252, 2021.

[Mozaffari *et al.*, 2020] S. Mozaffari, O. Y. Al-Jarrah, M. Dianati, P. Jennings, and A. Mouzakitis, Deep learning-based vehicle behavior prediction for autonomous driving applications: A review, *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[Wiest *et al.*, 2012] J. Wiest, M. Höffken, U. Kreßel, and K. Dietmayer, Probabilistic trajectory prediction with Gaussian mixture models, in *2012 IEEE Intelligent Vehicles Symposium*, 2012: IEEE, pp. 141-146.

[Deo *et al.*, 2018] N. Deo, A. Rangesh, and M. M. Trivedi, How would surround vehicles move? a unified framework for maneuver classification and motion prediction, *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 2, pp. 129-140, 2018.

[Yu *et al.*, 2021] D. Yu, H. Lee, T. Kim, and S.-H. Hwang, Vehicle trajectory prediction with lane stream attention-based LSTMs and road geometry linearization, *Sensors*, vol. 21, no. 23, p. 8152, 2021.

[Deo and Trivedi, 2018] N. Deo and M. M. Trivedi, Multi-modal trajectory prediction of surrounding vehicles with maneuver based lstms, in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018: IEEE, pp. 1179-1184.

[Han *et al.*, 2019] T. Han, J. Jing, and Ü. Özgüner, Driving intention recognition and lane change prediction on the highway, in *2019 IEEE Intelligent Vehicles Symposium (IV)*, 2019: IEEE, pp. 957-962.

[Nikhil and Morris, 2018] N. Nikhil and B. Tran Morris, Convolutional neural network for trajectory prediction, in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018, pp. 0-0.

[Lee *et al.*, 2017] D. Lee, Y. P. Kwon, S. McMains, and J. K. Hedrick, Convolution neural network-based lane change intention prediction of surrounding vehicles for ACC, in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 2017: IEEE, pp. 1-6.

[Tang *et al.*, 2008] T. Tang, H. Huang, S. Wong, and R. Jiang, A car-following model with the anticipation effect of potential lane changing, *Acta Mechanica Sinica*, vol. 24, no. 4, pp. 399-407, 2008.

[Yuan *et al.*, 2018] W. Yuan, Z. Li, and C. Wang, Lane-change prediction method for adaptive cruise control system with hidden Markov model, *Advances in Mechanical Engineering,* vol. 10, no. 9, p. 1687814018802932, 2018.

[Krajewski *et al.*, 2018] R. Krajewski, J. Bock, L. Kloeker, and L. Eckstein, The Highd dataset: A drone dataset of naturalistic vehicle trajectories on German highways for validation of highly automated driving systems, in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018: IEEE, pp. 2118-2125.

[Colyar and Halkias, 2006] J. Colyar and J. Halkias, Us highway 80 dataset, federal highway administration (fhwa), vol, *Tech, no. Rep,* 2006.

[Colyar and Halkias., 2007] J. Colyar and J. Halkias, US highway 101 dataset, *Federal Highway Administration (FHWA), Tech. Rep. FHWA-HRT-07-030*, pp. 27-69, 2007.

[Sutskever *et al.*, 2014] Sutskever, I., Vinyals, O., & Le, Q. V. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27, 2014

[Malinauskas, 2014] R. M. Malinauskas, The intelligent driver model: Analysis and application to adaptive cruise control, *Clemson University,* 2014.