

Received November 11, 2021, accepted December 21, 2021, date of publication December 24, 2021, date of current version December 31, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3138502

Driving Style-Based Conditional Variational Autoencoder for Prediction of Ego Vehicle Trajectory

DONGCHAN KIM¹, HYUKJU SHON², NAHYUN KWEON³, SEUNGWON CHOI²,
CHANUK YANG¹, AND KUNSOO HUH¹, (Member, IEEE)

¹Department of Automotive Engineering, Hanyang University, Seoul 04763, Republic of Korea

²Department of Automotive Engineering (Automotive-Computer Convergence), Hanyang University, Seoul 04763, Republic of Korea

³Department of Automotive Electronic Control Engineering, Hanyang University, Seoul 04763, Republic of Korea

Corresponding author: Kunsoo Huh (khuh2@hanyang.ac.kr)

This work was supported by the Ministry of Trade, Industry, and Energy (MOTIE), South Korea, through the Technology Innovation Program (Industrial Strategic Technology Development Program, Development of Test Procedure Standards for V2I Connected Automated Driving Systems) under Grant 20014460.

ABSTRACT Trajectory prediction of the ego vehicle is essential for advanced driver assistance systems to function properly. By recognizing various driving styles and predicting trajectories reflecting them, the prediction performance is enhanced, and a personalized trajectory can be generated. Therefore, we propose to combine driving style recognition and trajectory prediction tasks using only in-vehicle CAN-bus sensor data for possible application to normal vehicles. The DeepConvLstm network was utilized for driving style recognition, and a generative-based model was used for trajectory prediction. The classified driving style was added as a condition to the network. In addition, the past trajectory of the ego vehicle is estimated and utilized as an additional input for performance improvement. The performance of the proposed method is analyzed in terms of the root mean squared error (RMSE) and mean absolute error (MAE) compared with the case wherein the driving style and the past trajectory are not conditioned or given, respectively. The results demonstrate the effectiveness of the proposed method.

INDEX TERMS Driving style recognition, trajectory prediction, conditional variational autoencoder, CAN data, Hardware-in-the-Loop simulation.

I. INTRODUCTION

Trajectory prediction of the ego vehicle is a key task for many advanced driver assistance systems (ADAS) that implement evasive maneuvers to avoid or mitigate oncoming hazards. However, as a wide variety of drivers exist, they have their own driving styles, which makes accurate prediction of a personalized trajectory challenging. Therefore, to predict the trajectory accurately, it is necessary to grasp the driving style and make a prediction that reflects it. Driving style recognition and trajectory prediction of ego vehicle have been widely investigated.

Several studies have been conducted to determine the driving styles of drivers. Recognition of the driving style has been formulated as a classification problem that can be solved by training neural networks. Streiffer *et al.* proposed

The associate editor coordinating the review of this manuscript and approving it for publication was Ehab Elattar¹.

Darnet to recognize whether the driver is distracted using the driver's facial camera and inertial measurement unit (IMU) sensor [1]. Lee *et al.* used thermal and infrared images to recognize the driving styles [2]. Dong *et al.* proposed ARNet to encode driving styles using a GPS sensor [3]. Constantinescu *et al.* investigated the modeling of the personal driving styles of various vehicle drivers using GPS data [4]. However many vehicles do not contain cameras capable of providing driver's facial information or GPS that can provide an accurate vehicle pose. Nevertheless, Shahverdy *et al.* expressed controller area network (CAN)-bus signals as an image and used a convolutional neural network (CNN) to classify the driver's behavior [5]. In addition, the deep convolutional long short-term memory (DeepConvLSTM) network was developed to recognize human activity [6]. This model is capable of learning features from multi-sensor data from the CAN-bus with their temporal information.

Several researchers have made considerable effort to predict vehicle trajectories [7]–[10]. Baumann *et al.* predicted the trajectory of the ego vehicle using a deep neural network, but lidar is used as an input to the network [7]. Feng *et al.* [8] proposed a trajectory prediction network using a conditional variational autoencoder (CVAE) approach. They improved the performance with a simple intention recognizer network as an additional model to the previous network. However, they used Next Generation Simulation (NGSIM) data [11], which can be obtained using GPS, radar, or lidar. Malla *et al.* [9] proposed a trajectory prediction network that considers the surrounding obstacles, including the motion of pedestrians, and the feature of this study uses video information. Sadeghian *et al.* [10] proposed path prediction using an attentive generative adversarial network (GAN) using the information of images and surrounding vehicles.

Conversely, attempts have been made to estimate the driving style and predict the trajectory simultaneously. Xing *et al.* proposed a driving style-based trajectory prediction of a leading vehicle [12]. They used a Gaussian mixture model (GMM) to distinguish the driving styles in an unsupervised manner and used different LSTM networks to predict the trajectory based on the driving styles. However, only longitudinal motion was predicted, and NGSIM data were utilized. Liu *et al.* estimated the driving style using a dynamic Bayesian network and predicted the trajectory using a Gaussian process model [13], [14]. The limitation is that they used a naturalistic vehicle trajectory data set called highD [15] which is similar to NGSIM.

In this paper, we propose an integrated method of driving style recognition and trajectory prediction to produce a personalized trajectory using only in-vehicle CAN-bus data. First, a total of three driving styles were classified, including normal, aggressive, and distracted. Subsequently, the recognized driving style-based trajectory prediction was proceeds. The contributions of this study can be summarized as follows:

- Predict the driving style-based multi-modal trajectory of the ego vehicle using a deep generative model called CVAE.
- Utilize only the in-vehicle CAN-bus sensor data for inference.
- Estimate the past trajectory to feed the network for improvement of the prediction performance.
- Use Hardware-in-the-Loop simulation (HILS) for data acquisition of various driving styles.

II. APPROACH

In this section, the main algorithms regarding to driving style recognition and trajectory prediction are explained. We also describe the estimation part of the past trajectory for improvement of prediction performance.

A. DRIVING STYLE RECOGNITION

The classifier network used in our work for driving style recognition is DeepConvLSTM network with a similar

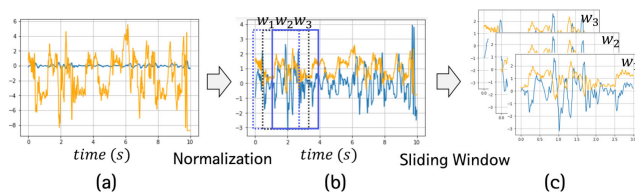


FIGURE 1. An example of CAN data processing. (a) The raw CAN data used in training phase (steering angle: blue, steering angle velocity: orange), shows large deviation on the data scale. (b) Normalized data. The deviation of the data scale decreases. (c) Sliding windows are stacked like an image.

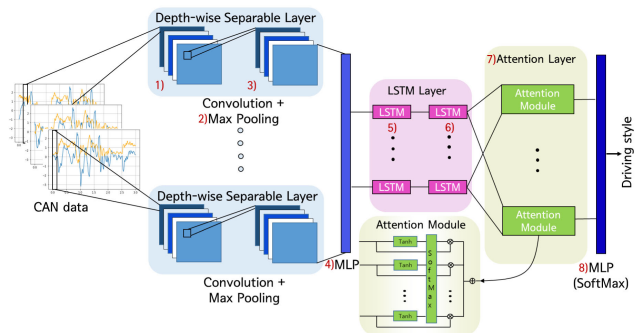


FIGURE 2. Driving style recognition network architecture.

architecture in [6]. To increase the performance of the network, the depth-wise separable network was used in the convolutional layer and attention layer was added [16], [17]. Additionally, the CAN data was pre-processed by a sliding window method used in [16]. The CAN data used consists of [brake, accel pedal, steering angle, steering angle rate, v_x , a_x , a_y , $\dot{\psi}$] with a sampling time of 0.1s for 5s where each element indicates brake pedal, accel pedal, steering wheel angle, steering wheel angle rate, longitudinal velocity, longitudinal acceleration, lateral acceleration and yawrate, respectively.

1) CAN DATA PROCESSING

Before applying the sliding window method, the data is normalized along each data scale.

$$X_i^n = \frac{X_i^n - \text{mean}(X^n)}{\text{std}(X^n)}, \quad (n = 1, \dots, D_n, i = 1, \dots, T) \quad (1)$$

where *mean* and *std* indicate the mean value and standard deviation, respectively. D_n denotes the number of data points used, and T denotes the time step of the data.

The normalized data (\mathcal{X}) is then cropped into the window, and the window is slid along the data by the step size. The total number of windows is stacked channel-wise to transform the data into an image form. The details of this process are presented in Fig.1. The processed data has the following structure: $W_x \times D_n \times W_n$ (W_x : window size, D_n : number of data, and W_n : number of windows).

2) NETWORK ARCHITECTURE

Our network is mainly composed of a convolutional layer, recurrent layer and attention layer. The convolutional layer

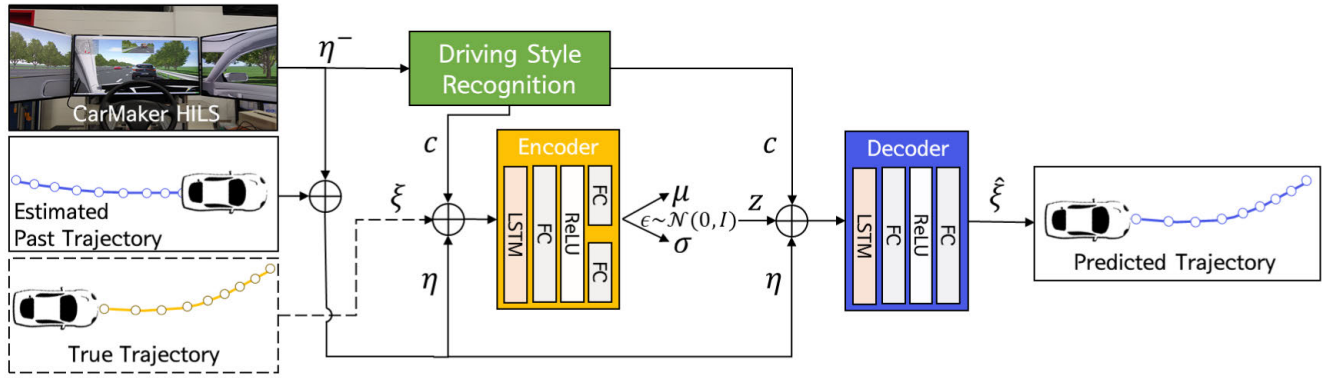


FIGURE 3. Proposed driving style-based CVAE.

extracts the features in the time series, and the recurrent layer learns the temporal information. The convolutional layer is modified using a depth-wise separable layer and max-pooling layer [16]. Unlike in [16], we used batch normalization instead of dropout to accelerate network training speed. The modified layer makes the network lighter and makes feature learning more efficient. We used LSTM for the recurrent layer to avoid the long-term dependency problem. The attention layer captures the relationship between each feature and is capable of learning the importance of the feature.

The overall network structure is shown in Fig. 2. First, the feature map is extracted from the convolutional layers. The extracted features are then passed through the LSTM layer, the attention layer, and the classifier layer, which is the fully-connected layer.

B. TRAJECTORY PREDICTION

The trajectory prediction method in our study is based on the CVAE structure which includes an encoder-decoder network with conditional inputs. Two types of conditional inputs are used in the proposed method. One is the driving style explained in Section II-A. The other is an embedded vector that is made in combination with CAN-bus data from Car-Maker HILS and the estimated past trajectory. The overall architecture is shown in Fig. 3. The true trajectory ξ , depicted with a dotted line, is used only during the training phase.

The CVAE consists of a generative model $p_\rho(\xi|\eta, c, z)$ and an inference model $q_\phi(z|\eta, c, \xi)$, and the latent variable z is expressed as follows using the reparameterization trick [18]:

$$z = \mu_\phi(\eta, c, \xi) + \sigma_\phi(\eta, c, \xi) \times \epsilon \tag{2}$$

where ϕ and ρ are the parameters of the encoder and decoder network, respectively. ϵ is sampled from a normal distribution $\mathcal{N}(0, I)$. To minimize the error between the predicted trajectory $\hat{\xi}$ and ground truth ξ , the reconstruction loss is defined as L2 loss. The entire CVAE network is trained by minimizing the loss function, defined as follows:

$$\begin{aligned} \mathcal{L} &= L_{REC} + L_{KL} \\ &= \|\xi - \hat{\xi}\|^2 + WD_{KL}(q_\phi(z|\eta, c, \xi)||p(z)) \end{aligned} \tag{3}$$

where W is a hyperparameter that balances the two losses. The former part represents the reconstruction loss, and the latter part represents the Kullback–Leibler (KL) divergence loss between the multivariate normal distribution and the output distribution from the encoder.

In the test phase, the encoding of the true future trajectory is not available; thus, we directly sample from $\mathcal{N}(0, I)$ as z , and only the decoder is used to obtain the predicted trajectory.

Different trajectories of driving styles can be generated by feeding different conditions. The driving style recognition network in Fig. 3 provides a probability output of the driving style prediction. Then, the maximum probability output, in one-hot vector form, is utilized in the encoder and decoder of the proposed CVAE architecture. For example, if [1, 0, 0] is fed, the driving style named normal is conditioned. Likewise, [0, 1, 0] for aggressive and [0, 0, 1] for distracted. Therefore, in the following section, we analyze the effects of this factor to determine whether different conditions result in different types of trajectories.

1) PAST TRAJECTORY ESTIMATION

The constant turn rate and acceleration model (CTRA) [19] was used to estimate the past trajectory. The CTRA model assumes that the turn rate and acceleration are constant. The state vector is expressed as follows:

$$\mathbf{x} = (x, y, \theta, v, a, \dot{\psi})^T \tag{4}$$

where x, y indicates the position of the vehicle, θ is the heading angle of the vehicle, v is the velocity, a is the acceleration, and $\dot{\psi}$ is the yawrate.

The state of the next time step is expressed by

$$\begin{aligned} x_{k+1} &= x_k + \int_0^{\Delta t} v_{x,k}(\tau) d\tau \\ &= x_k + \frac{(v_k + a_k \Delta t) \sin(\dot{\psi}_k \Delta t + \theta_k) - v_k \sin \theta_k}{\dot{\psi}_k} \\ &\quad + a_k \frac{\cos(\dot{\psi}_k \Delta t + \theta_k) - \cos \theta_k}{\dot{\psi}_k^2} \end{aligned} \tag{5}$$

$$\begin{aligned}
 y_{k+1} &= y_k + \int_0^{\Delta t} v_{y,k}(\tau) d\tau \\
 &= y_k + \frac{v_k \cos \theta_k - (v_k + a_k \Delta t) \cos(\dot{\psi}_k \Delta t + \theta_k)}{\dot{\psi}_k} \\
 &\quad + a_k \frac{\sin(\dot{\psi}_k \Delta t + \theta_k) - \sin \theta_k}{\dot{\psi}_k^2} \quad (6)
 \end{aligned}$$

$$\theta_{k+1} = \theta_k + \dot{\psi}_k \Delta t \quad (7)$$

$$v_{k+1} = v_k + a_k \Delta t \quad (8)$$

$$a_{k+1} = a_k \quad (9)$$

$$\dot{\psi}_{k+1} = \dot{\psi}_k \quad (10)$$

where the subscript k is the time step, and Δt is the prediction time interval.

The state-space equation considering the process noise is expressed as follows:

$$\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k) + \mathbf{w}_k \quad (11)$$

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{r}_k \quad (12)$$

$$\mathbf{w}_k = [w_x, w_y, w_\theta, w_v, w_a, w_{\dot{\psi}}]^T \quad (13)$$

$$\mathbf{r}_k = [r_v, r_a, r_{\dot{\psi}}]^T \quad (14)$$

where \mathbf{w}_k and \mathbf{r}_k respectively indicate system noise and observation noise defined as the Gaussian distribution.

Unscented Kalman filter (UKF) was utilized in our study [20] to solve nonlinear problems. To consider the uncertainty in the nonlinear dynamic model, the unscented transform (UT) was used. A fixed number of sigma points were selected from the original distribution to estimate the transformed distribution. The UT process was used with the motion model to perform the trajectory estimation task by considering the uncertainty in (11).

The past trajectory initialized with $(x_1 = 0, y_1 = 0, \theta_1 = 0)$ is estimated during the history horizon as shown in Fig. 3a. The in-vehicle sensor data composed of the longitudinal velocity, v_x , the longitudinal acceleration, a_x , and yawrate, $\dot{\psi}$ was used as measurement in UKF. Then the estimated trajectory was transformed to make the last estimate, which is at the current time step to $(x_{T_{obs}} = 0, y_{T_{obs}} = 0, \theta_{T_{obs}} = 0)$ as shown in Fig. 3b. This is because the prediction is performed in the local coordinate system of the ego vehicle.

III. EXPERIMENTS

In this section, we collected a dataset using CarMaker HILS by IPG Automotive shown in Fig. 5. The implementation details of the driving style recognition network and the trajectory prediction network are described.

A. DATASET ACQUISITION

In order to examine the proposed method, we carried out multiple sets of driving through the CarMaker HILS for data acquisition. Highway scenario in the 3-lane road is used and the surrounding vehicles were randomly spawned. Straight roads and curved roads with different curvatures were designed for data collection. The data collected consisted of [brake, accel pedal, steering angle, steering angle rate, v_x ,

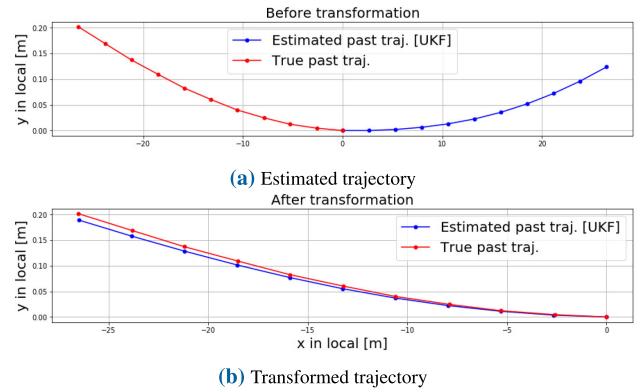


FIGURE 4. An example of the past trajectory estimation.

$a_x, a_y, \dot{\psi}$] with a sampling time of 0.1s, and it is depicted as η^- in Fig. 3.

Twelve drivers participated in the experiments. They were asked to drive in all types of driving styles, which consisted of normal, aggressive, and distracted.

A Normal driving style is defined as driving safely at a speed similar to the traffic flow, changing lanes smoothly, accelerating and decelerating slowly, and not overtaking the preceding vehicles often. An Aggressive driving style is defined as driving at a faster speed than the traffic flow, changing lanes abruptly and hazardously, accelerating and decelerating rapidly, and overtaking the preceding vehicles. Finally, to create a distracted driving situation, the drivers were asked to text messages on their mobile phones or watch videos while driving.

In Figs. 6 and 7, three different driving styles are visualized as examples based on arbitrarily selected CAN-bus data. The blue, yellow, and red dots represent the distributions of normal, distracted, and aggressive driving styles, respectively. As can be observed in the two figures, different distributions are shown based on the three driving styles. The distribution of the distracted driving style is similar to that of the normal driving style, but wider because unstable driving was performed by the drivers.

B. IMPLEMENTATION DETAILS

1) DRIVING STYLE RECOGNITION NETWORK

We pre-processed the CAN data using a window size of 30, step size of 5, and window number of 4. After the pre-processing step, the network input size was $30 \times 9 \times 4$. The network is composed of two depth-wise separable convolutional layers with one max pooling layer. We used the rectified linear unit (ReLU) activation function after each convolutional layer and batch normalization for regularization. In addition, two LSTM layers were used. The details of the network are listed in Table. 1. The numbers 1) through 8) indicate the process represented in Fig. 2. In the training process, cross-entropy was used for the loss function. The optimization was performed using a standard Adam optimizer with a learning rate of 0.001.



FIGURE 5. IPG Automotive / CarMaker HILS.

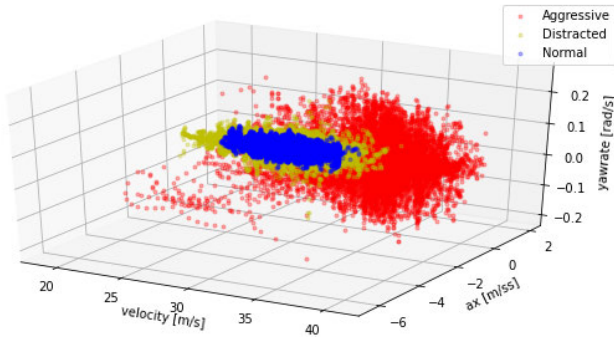


FIGURE 6. Visualization of the distributions for three driving styles with respect to velocity, longitudinal acceleration and yawrate.

2) TRAJECTORY PREDICTION NETWORK

As shown in Fig. 3, the proposed architecture consists of a driving style recognition network, encoder, and decoder. In the encoder, the concatenated vector of η , c , and ξ is fed into an LSTM layer with a 64-dimensional cell state. Then, the result passes through a fully-connected layer with 64 hidden units followed by an activation function and two fully-connected layers with eight hidden units, which is the dimension of the latent variable. In the decoder, the concatenated vector of η , c , and z first passes through an LSTM layer with 64-dimensional cell state followed by a fully-connected layer with 64 hidden units. Then the output passes through an activation function and a fully-connected layer with 20 hidden units which is the dimension of the predicted trajectory. *ReLU* is used for the activation function. The sampling time is 0.1s, and we predicted the next 1s using the last 1s information. W in Eq. 3 was set to 0.001 in our study. The optimization was performed using an Adam optimizer with a learning rate of 0.001.

IV. RESULTS

In this section, first, the results of driving style recognition are explained, and then the results of trajectory prediction are described. Subsequently, the results are compared to other baselines, and the performance analysis is conducted.

A. DRIVING STYLE RECOGNITION

The performance of the network was tested using the testdata set obtained from HILS. The accuracy of the network is calculated at all time steps using the same window size and window number as the training. The maximum probability output is the predicted driving style of the network. If the

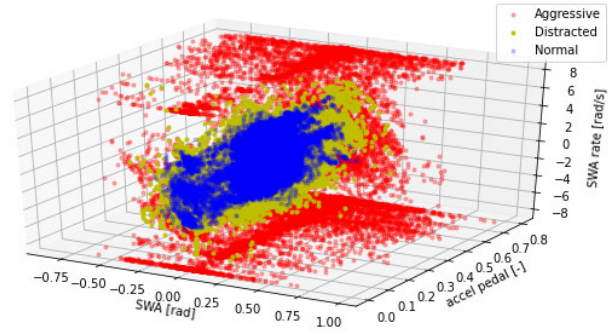


FIGURE 7. Visualization of the distributions for three driving styles with respect to steering wheel angle, accel pedal and steering wheel angle rate.

TABLE 1. Driving style recognition network parameters.

| Layer | Layer Information |
|--|---|
| 1) CNN layer (Depth-wise Separable layer) | Output Channel : 128 Kernel Size : (1×7) |
| 2) Max pooling layer | Kernel Size: (1×5) |
| 3) CNN layer (Depth-wise Separable layer) | Output Channel : 64 Kernel Size : (1×5) |
| 4) MLP layer | Output Feature : 4×9 |
| 5) LSTM layer | Hidden States : 128 |
| 6) LSTM layer | Hidden States : 128 |
| 7) Attention layer | Output Feature : 9 |
| 8) MLP layer | Output Size : 3 |

TABLE 2. Driving style recognition network performance.

| | Normal | Aggressive | Distracted | Total |
|--------------|--------|------------|------------|-------|
| Accuracy [%] | 96.7 | 99.8 | 95.5 | 95.5 |

prediction style matches the true driving style of the data, it is evaluated as correct. Table. 2 lists the network accuracy for predicting each driving style.

B. TRAJECTORY PREDICTION

1) METRIC

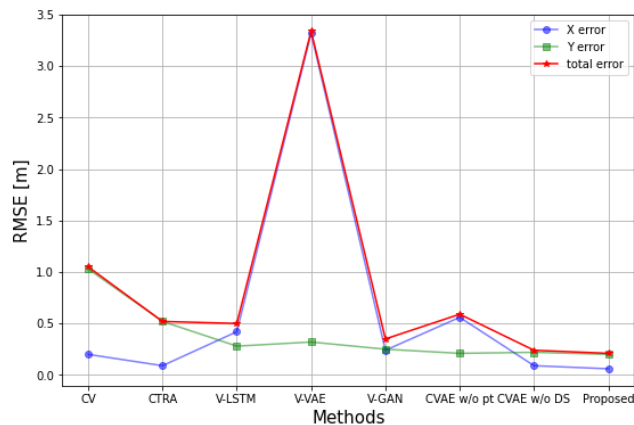
In our study, we used the root mean square error (RMSE) and mean absolute error (MAE) as the evaluation metrics which are calculated as follows:

$$RMSE_X = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \hat{x}_i)^2} \quad (15)$$

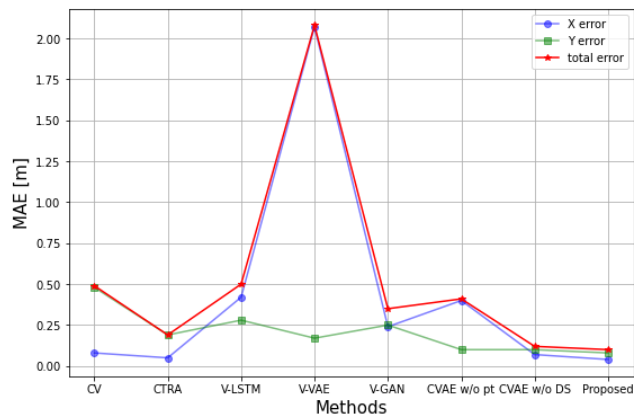
$$RMSE_Y = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2} \quad (16)$$

$$MAE_X = \frac{1}{N} \sum_{i=1}^N |x_i - \hat{x}_i| \quad (17)$$

$$MAE_Y = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i| \quad (18)$$



(a) Performance comparison by RMSE.



(b) Performance comparison by MAE.

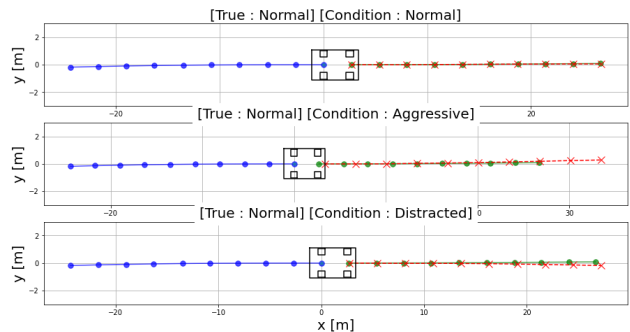
FIGURE 8. Performance comparison by RMSE and MAE.

where N is the total number of prediction steps, \hat{x}_i and \hat{y}_i are the longitudinal and lateral values of the predicted trajectory, respectively, and x_i and y_i are the longitudinal and lateral values of the true trajectory, respectively.

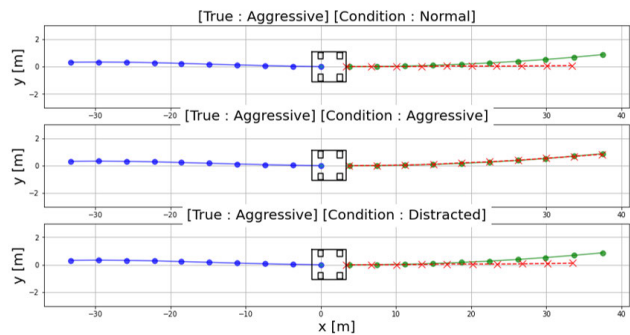
2) COMPARED MODELS

In the experiment, we compared eight different models in our work as follows.

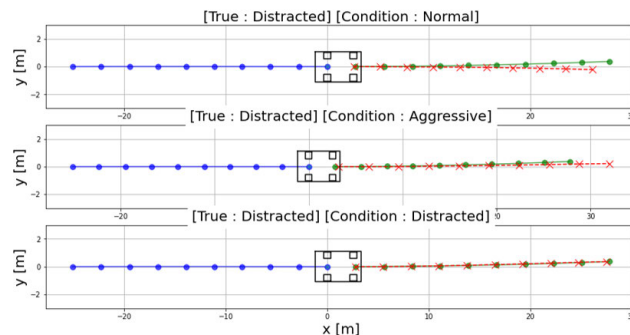
- **Constant velocity (CV) model** : We implemented the UKF using the CV model for the non-deep learning baseline. The state vector is expressed as $\mathbf{x} = (x, y, \theta, v)^T$.
- **CTRA model** : We implemented the UKF using the CTRA model for another non-deep learning baseline. The same model is used as explained in section II-B1.
- **Vanilla LSTM model (V-LSTM)** : The vanilla LSTM model is used to predict the future trajectory. The past CAN-bus data is fed to an LSTM. The network includes 1 LSTM layer with a 64-dimensional cell state followed by a fully-connected layer.
- **Vanilla VAE (V-VAE)** : The vanilla VAE was used to predict the future trajectory. The true future trajectory was fed to an encoder in the training phase. The model



(a) True condition : normal



(b) True condition : aggressive



(c) True condition : distracted

FIGURE 9. Prediction results given different driving styles.

structure is identical to Fig. 3 except for the conditions η and c where the details in parameter setting are explained in Section III-B2.

- **Vanilla Generative Adversarial Network (V-GAN)** : The vanilla GAN was used to predict the future trajectory. The true future trajectory was fed to an encoder in the training phase. The model structure is composed of encoder, generator, and discriminator. It consists of 2 LSTM layers with a 32-dimensional cell state followed by 3 fully-connected layers. The hidden sizes of fully-connected layers of encoder, generator and discriminator are (64, 160, 320), (160, 80, 20), (16, 8, 1), respectively.
- **Conditional Variational Autoencoder (CVAE without past trajectory)** : The CVAE model without the associated past trajectory is also used for comparison.

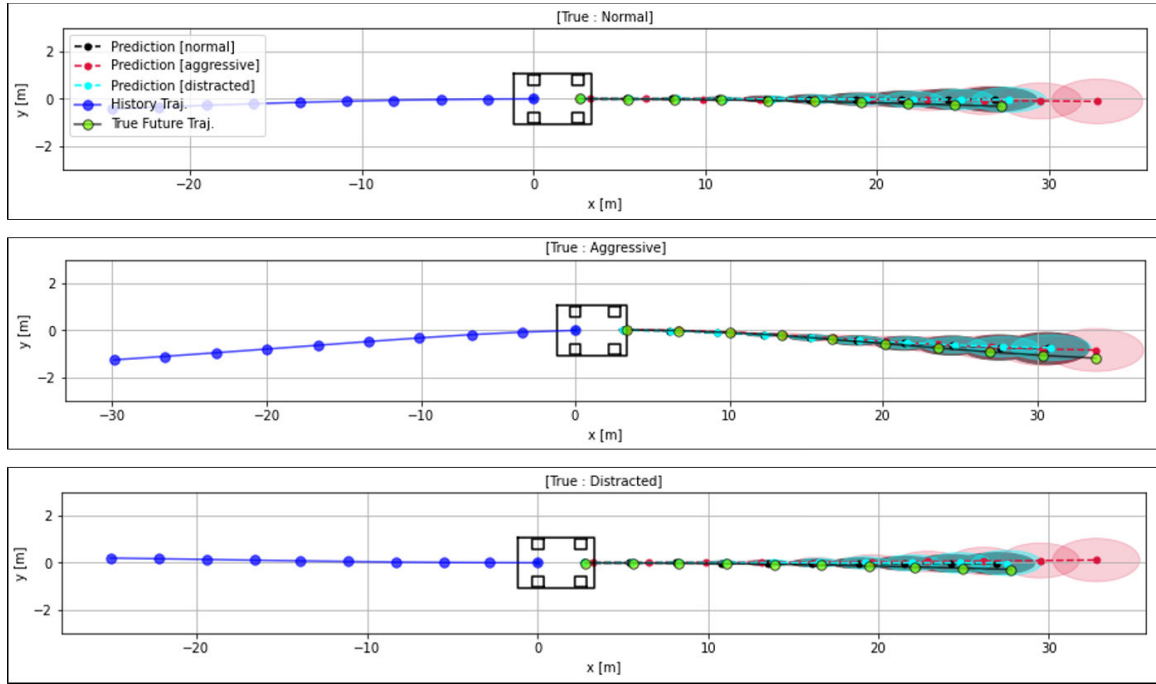


FIGURE 10. All distribution with the proposed CVAE: The line with blue circle denotes the history trajectory and the line with green circle denotes the future trajectory. The dotted lines with ellipses indicate the prediction distributions given all driving styles.

TABLE 3. Performance comparison by RMSE.

| | X error [m] | Y error [m] | total error [m] |
|---------------------|-------------|-------------|-----------------|
| CV | 0.20 | 1.03 | 1.05 |
| CTRA | 0.09 | 0.52 | 0.52 |
| V-LSTM | 0.42 | 0.28 | 0.50 |
| V-VAE | 3.32 | 0.32 | 3.34 |
| V-GAN | 0.24 | 0.25 | 0.35 |
| CVAE w/o past traj. | 0.56 | 0.21 | 0.59 |
| CVAE w/o DS | 0.09 | 0.22 | 0.24 |
| Proposed | 0.06 | 0.20 | 0.21 |

TABLE 4. Performance comparison by MAE.

| | X error [m] | Y error [m] | total error [m] |
|---------------------|-------------|-------------|-----------------|
| CV | 0.08 | 0.48 | 0.49 |
| CTRA | 0.05 | 0.19 | 0.19 |
| V-LSTM | 0.27 | 0.17 | 0.32 |
| V-VAE | 2.07 | 0.17 | 2.08 |
| V-GAN | 0.15 | 0.10 | 0.19 |
| CVAE w/o past traj. | 0.40 | 0.10 | 0.41 |
| CVAE w/o DS | 0.07 | 0.10 | 0.12 |
| Proposed | 0.04 | 0.08 | 0.10 |

The model structure is identical to Fig. 3 except for the past trajectory information.

- **Conditional Variational Autoencoder (CVAE w/o driving style)** : The CVAE model without driving style condition was used for comparison. The model structure is identical to Fig. 3 except for the condition, c , from the driving style recognition network.
- **Proposed model** : The proposed model uses driving style as an additional condition for the CVAE architecture. The estimated past trajectory is also used as network input.

3) PERFORMANCE EVALUATION

In this section, tables and figures are presented and explained to analyze the performance. Tables 3 and 4 show the longitudinal and lateral position errors in terms of the RMSE

and MAE of 1s, respectively. The Fig. 8 shows the RMSE and MAE as graphs for clear comparison. The first non-deep learning baseline that uses the CV model with the UKF has significant errors, especially in the lateral position. The other non-deep learning model that uses CTRA has good estimation results in the longitudinal position. However, despite the improvement compared with the CV model, it shows a large error in the lateral position. V-VAE showed improvement in the lateral position compared with the baseline models, but it performed poorly in the longitudinal position estimation. CVAE, which has no past trajectory as an additional input, has a larger error in the longitudinal direction compared with the baselines. However, the estimation performance in terms of the lateral direction increased. In addition, it shows a dramatic improvement compared with the V-VAE in terms of the lateral position. CVAE without driving style condition but with past

trajectory generally has better performance than previous methods. Our proposed model makes progress over the above methods and achieves the best results. This result shows that when the estimated past trajectory is conditioned, the trajectory estimation performance is significantly improved. Furthermore, using the driving style condition improves the estimation performance and allows multi-modal prediction.

Fig. 9 shows the prediction results for the test dataset. The blue line represents the past trajectory, the green line represents the true future trajectory, and the red dotted line represents the prediction. Fig. 9a shows the result when the true driving style is normal, and each driving style is fed into the model as a condition. The results show that the prediction performance improves when the correct driving style is conditioned. In addition, when aggressive is conditioned, the trajectory tends to go further. In Fig. 9b and 9c, the trajectory is predicted accurately when the true driving style is conditioned. It also indicates that the model tends to produce a shorter trajectory under normal conditions.

In Fig. 10, all the trajectory distributions for the three driving styles are depicted. The results show that our proposed method can generate multi-modal predictions by assigning the probability of each driving style obtained from the recognition network. The trajectory distribution from each modal was drawn along with an ellipse corresponding to 3σ . The prediction distribution with the true condition has a better prediction performance than the others. Additionally, the distribution of normal condition (black) tends to lie in a shorter area than the others, and the distribution of aggressive conditions (red) tends to lie further. To quantitatively evaluate the performance of each algorithm in this specific test case, the prediction results in terms of total MAE are shown as follows:

- **True label - normal** : CV : 1.03m, CTRA : 0.29m, V-LSTM : 0.54m, V-VAE : 3.23m, V-GAN : 0.17m, CVAE w/o past traj. : 1.19m, CVAE w/o DS : 0.20m, Proposed : 0.05m
- **True label - aggressive** : CV : 5.55m, CTRA : 2.07m, V-LSTM : 0.22m, V-VAE : 7.35m, V-GAN : 0.21m, CVAE w/o past traj. : 0.34 m, CVAE w/o DS : 0.27m, Proposed : 0.07m
- **True label - distracted** : CV : 0.49m, CTRA : 2.08m, V-LSTM : 0.19m, V-VAE : 4.14m, V-GAN : 0.14m, CVAE w/o past traj. : 0.70m, CVAE w/o DS : 0.21m, Proposed : 0.05m

V. CONCLUSION

In this paper, a driving style-based prediction model is proposed for future trajectory prediction of the ego vehicle using only the in-vehicle CAN-bus data. The proposed model uses a DeepConvLSTM network with a sliding window approach for driving style recognition. Then, the classified driving style is assigned to the CVAE structure with the estimated past trajectory. The proposed method is trained based on the experimental data collected using HILS, and the evaluation

results show that the proposed method outperforms the base-line methods. In addition, the trained model can produce multi-modal predictions corresponding to driving styles with interpretability. Experiments in more extensive driving scenarios that consider a wider variety of driving styles and validation on the real-car dataset remain for future study.

REFERENCES

- [1] C. Streiffer, R. Raghavendra, T. Benson, and M. Srivatsa, "DarNet: A deep learning solution for distracted driving detection," in *Proc. 18th ACM/IFIP/USENIX Middleware Conf., Ind. Track*, 2017, pp. 22–28.
- [2] K. W. Lee, H. S. Yoon, J. M. Song, and K. R. Park, "Convolutional neural network-based classification of driver's emotion during aggressive and smooth driving using multi-modal camera sensors," *Sensors*, vol. 18, no. 4, p. 957, 2018.
- [3] W. Dong, T. Yuan, K. Yang, C. Li, and S. Zhang, "Autoencoder regularized network for driving style representation learning," 2017, *arXiv:1701.01272*.
- [4] Z. Constantinescu, C. Marinoiu, and M. Vladiou, "Driving style analysis using data mining techniques," *Int. J. Comput. Commun. Control*, vol. 5, no. 5, pp. 654–663, 2010.
- [5] M. Shahverd, M. Fathy, R. Berangi, and M. Sabokrou, "Driver behavior detection and classification using deep convolutional neural networks," *Expert Syst. Appl.*, vol. 149, Jul. 2020, Art. no. 113240.
- [6] F. J. Ordóñez and D. Roggen, "Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, 2016.
- [7] U. Baumann, C. Guiser, M. Herman, and J. M. Zollner, "Predicting ego-vehicle paths from environmental observations with a deep neural network," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 4709–4716.
- [8] X. Feng, Z. Cen, J. Hu, and Y. Zhang, "Vehicle trajectory prediction using intention-based conditional variational autoencoder," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 3514–3519.
- [9] S. Malla, B. Dariush, and C. Choi, "TITAN: Future forecast using action priors," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11186–11196.
- [10] A. Sadeghian, V. Kosaraju, A. Sadeghian, N. Hirose, H. Rezatofighi, and S. Savarese, "SoPhie: An attentive GAN for predicting paths compliant to social and physical constraints," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1349–1358.
- [11] J. Colyar and J. Halkias, "Us highway 101 dataset," Federal Highway Admin., Washington, DC, USA, Tech. Rep., FHWA-HRT-07-030, 2007, pp. 27–69.
- [12] Y. Xing, C. Lv, and D. Cao, "Personalized vehicle trajectory prediction based on joint time-series modeling for connected vehicles," *IEEE Trans. Veh. Technol.*, vol. 69, no. 2, pp. 1341–1352, Feb. 2020.
- [13] J. Liu, Y. Luo, H. Xiong, T. Wang, H. Huang, and Z. Zhong, "An integrated approach to probabilistic vehicle trajectory prediction via driver characteristic and intention estimation," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 3526–3532.
- [14] J. Liu, H. Xiong, T. Wang, H. Huang, Z. Zhong, and Y. Luo, "Probabilistic vehicle trajectory prediction via driver characteristic and intention estimation model under uncertainty," *Ind. Robot, Int. J. Robot. Res. Appl.*, vol. 48, no. 6, pp. 778–791, Nov. 2021.
- [15] 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 *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 2118–2125.
- [16] J. Zhang, Z. Wu, F. Li, C. X. T. Ren, J. Chen, and L. Liu, "A deep learning framework for driving behavior identification on in-vehicle CAN-BUS sensor data," *Sensors*, vol. 19, no. 6, p. 1356, Mar. 2019.
- [17] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1251–1258.
- [18] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, *arXiv:1312.6114*.
- [19] R. Schubert, E. Richter, and G. Wanielik, "Comparison and evaluation of advanced motion models for vehicle tracking," in *Proc. 11th Int. Conf. Inf. Fusion, Jun./Jul. 2008*, pp. 1–6.
- [20] E. A. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," in *Proc. IEEE Adapt. Syst. Signal Process., Commun., Control Symp.*, Oct. 2000, pp. 153–158.



DONGCHAN KIM received the B.S. degree in automotive engineering from Hanyang University, Seoul, Republic of Korea, in 2015. He is currently pursuing the Ph.D. degree with the Department of Automotive Engineering. His research interests include motion planning, trajectory prediction, vehicle state estimation, reinforcement learning, and deep learning.



SEUNGWON CHOI received the B.S. degree in automotive engineering from Hanyang University, Seoul, Republic of Korea, in 2019. He is currently pursuing the Ph.D. degree with the Department of Automotive Engineering. His research interests include trajectory prediction, motion planning, localization, and deep learning.



HYUKJU SHON received the B.S. degree in mechanical engineering from Hanyang University, Seoul, Republic of Korea, in 2018. He is currently pursuing the Ph.D. degree with the Department of Automotive Engineering. His research interests include sensor fusion, localization and mapping in autonomous vehicle, and deep learning-based path prediction.



CHANUK YANG received the B.S. degree in mechanical system and design engineering from Hongik University, Seoul, Republic of Korea, in 2016. He is currently pursuing the Ph.D. degree with the Department of Automotive Engineering. His research interests include vehicle state estimation, sensor fusion, reinforcement learning, and deep learning.



NAHYUN KWEON received the B.S. degree in automotive engineering from Hanyang University, Seoul, Republic of Korea, in 2020. He is currently pursuing the M.S. degree with the Department of Automotive Electronics and Control Engineering. His research interests include trajectory prediction, risk assessment, vehicle control, deep learning, and ADAS.



KUNSOO HUH (Member, IEEE) received the Ph.D. degree from the University of Michigan, Ann Arbor, MI, USA, in 1992. He is currently a Professor with the Department of Automotive Engineering, Hanyang University, Seoul, South Korea. His current research interests include machine monitoring and control, with emphasis on their applications to vehicular systems, sensor-based active safety systems, V2X-based safety systems, autonomous vehicle control, and AI applications in autonomous vehicle.

...