

Article

Vehicle Trajectory Prediction with Lane Stream Attention-Based LSTMs and Road Geometry Linearization

Dongyeon Yu , Honggyu Lee, Taehoon Kim and Sung-Ho Hwang * 

Department of Mechanical Engineering, Sungkyunkwan University, 2066 Seobu-ro, Suwon 16419, Korea; yunotions2@skku.edu (D.Y.); ghdrb9138@skku.edu (H.L.); th94228@skku.edu (T.K.)

* Correspondence: hsh0818@skku.edu

Abstract: It is essential for autonomous vehicles at level 3 or higher to have the ability to predict the trajectories of surrounding vehicles to safely and effectively plan and drive along trajectories in complex traffic situations. However, predicting the future behavior of vehicles is a challenging issue because traffic vehicles each have different drivers with different driving tendencies and intentions and they interact with each other. This paper presents a Long Short-Term Memory (LSTM) encoder–decoder model that utilizes an attention mechanism that focuses on certain information to predict vehicles’ trajectories. The proposed model was trained using the Highway Drone (HighD) dataset, which is a high-precision, large-scale traffic dataset. We also compared this model to previous studies. Our model effectively predicted future trajectories by using an attention mechanism to manage the importance of the driving flow of the target and adjacent vehicles and the target vehicle’s dynamics in each driving situation. Furthermore, this study presents a method of linearizing the road geometry such that the trajectory prediction model can be used in a variety of road environments. We verified that the road geometry linearization mechanism can improve the trajectory prediction model’s performance on various road environments in a virtual test-driving simulator constructed based on actual road data.

Keywords: trajectory prediction; LSTM encoder–decoder model; attention mechanism; road geometry; autonomous driving



Citation: Yu, D.; Lee, H.; Kim, T.; Hwang, S.-H. Vehicle Trajectory Prediction with Lane Stream Attention-Based LSTMs and Road Geometry Linearization. *Sensors* **2021**, *21*, 8152. <https://doi.org/10.3390/s21238152>

Academic Editors: Alexander Ferrein, Michael Reke and Stefan Schiffer

Received: 23 October 2021
Accepted: 4 December 2021
Published: 6 December 2021

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Intelligent vehicles, including partially automated vehicles that are equipped with Adaptive Cruise Control (ACC), require the ability to drive strategically according to the flow of traffic while simultaneously ensuring safety. Strategic driving includes determining when to change lanes between surrounding vehicles, passing low-speed or erratically behaving vehicles, and creating space for surrounding vehicles to change lanes. Autonomous vehicles must have the ability to predict the future behaviors and trajectories of surrounding vehicles to implement these features. Predicting surrounding vehicles’ behaviors is a core element that has a significant effect on everything from planning trajectories for basic driving [1,2] to high-level features such as predictive control for improving comfort and safety [3] and high fuel efficiency driving [4]. The ability to predict the future behavior of surrounding vehicles requires sensing technology that accurately recognizes obstacles [5] and interacts with surrounding autonomous driving systems [6].

To date, several methods for predicting vehicle trajectories have been proposed. The classic trajectory prediction method that is generally employed uses a Bayesian filtering technique such as a Kalman filter in the vehicle motion model [7–9]. These methods use simple models to ensure quick computation speed and are good at predicting the near future; however, they show poor performance regarding long-term predictions that reflect the nonlinear movements of vehicles. To address these limitations, more elaborate models such as the Gaussian mixture model [10] and Dynamic Bayesian Network (DBN) [11]

have been proposed. Nevertheless, they have not been sufficient for depicting the various nonlinear dynamic motions of actual vehicles. Laugier et al. proposed a method for predicting a future path using the Hidden Markov Model (HMM), which probabilistically models the change in a specific state [10,11]. Schreier et al. presented a method to improve long-term prediction performance by designing a Bayesian network to classify vehicle behavior and predict detailed routes [12].

Recent studies have used various deep learning methods for trajectory prediction. These studies have mainly used the Recurrent Neural Network (RNN) technique to learn and predict time series data [12]. The challenge of predicting vehicle trajectories, in which a vehicle's future position is predicted based on a series of past data, has aspects in common with the work on voice recognition and Natural Language Processing (NLP), which have garnered success in the field of machine learning. The Long Short-Term Memory (LSTM) model, which resolves the basic RNN model's vanishing gradient challenge, has exhibited excellent performance in the field of time series data learning [13–16]. Studies have been presented that have improved long-term prediction by applying the benefits of the LSTM model to the problem of sequence-to-sequence prediction [17] of vehicle trajectories [18–21] or by applying them to dynamic obstacles such as pedestrians [22].

This study proposes methods to improve the vehicle future trajectory prediction model's learning efficiency and long-term prediction performance. The driving conditions of surrounding vehicles were transformed from the perspective of the predicted target vehicle, and an attention mechanism was applied to the LSTM model to selectively focus on important information. In order to apply the model developed from the traffic dataset to the natural environment, we propose a method of transforming a complex road shape into a simplified straight frame. The framework of the proposed vehicle trajectory prediction method is illustrated in Figure 1, and it comprises the two parts below.

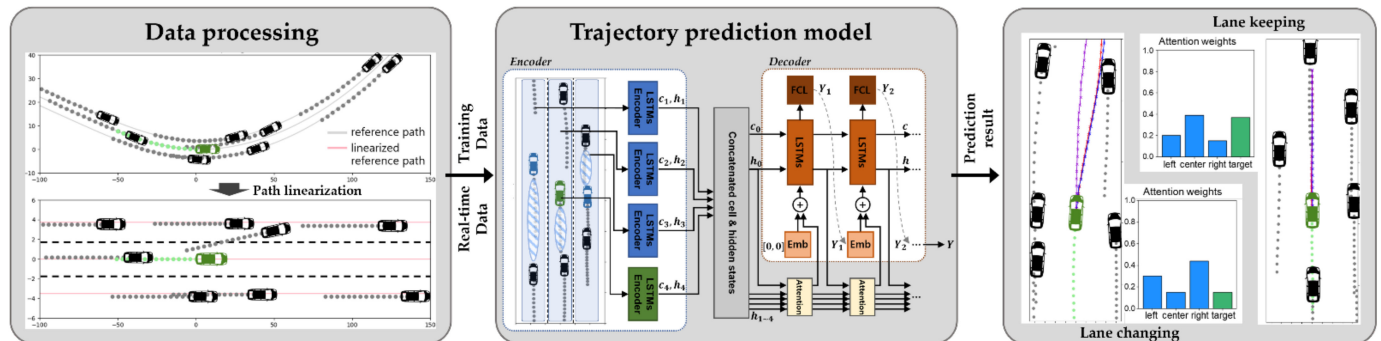


Figure 1. Framework of the proposed vehicle trajectory prediction method.

Section 3 presents a lane stream attention-based LSTM encoder–decoder model. This inputs information related to surrounding vehicles and outputs the future coordinates of the target vehicles. It is based on a local coordinate system that is fixed on the rear wheel surface center of the target vehicle. This method does not simply input all the position information related to surrounding vehicles in a specified area around the target vehicle; instead, it summarizes the information as data that depict the adjacent lane's traffic stream information and the target vehicle's main status. It is possible to improve learning efficiency and long-term prediction performance by configuring an attention mechanism that can focus on each situation among the context vectors of each lane and target vehicle state. This study used the Highway Drone (HighD) dataset [23], which is a large-scale, naturalistic traffic vehicle trajectory dataset, to train the model.

Section 4 presents a road geometry linearization method for effectively using the trajectory prediction model on roads with various geometries. Most previous studies have developed trajectory prediction models based on data obtained from straight roads and then evaluated the performance on these road sections. Therefore, it is unlikely that the reported performance will be achieved when using these models in autonomous vehicles

that must drive on real roads with various geometries. A road geometry linearization method was developed by noting that actual drivers determine their driving intentions based on the longitudinal or lateral positions of surrounding vehicles within each lane rather than on the relative positions of surrounding vehicles. This study verifies that it is possible to obtain sufficient trajectory prediction performance, even on curved roads, when the proposed linearization method is used.

2. Related Work

The various proposed vehicle trajectory prediction methods can be classified according to the interactions between the surrounding vehicles and the target vehicle [20,24]. Recent related studies using deep learning models to effectively improve long-term prediction performance are summarized as follows.

Independent prediction: Initial studies on vehicle trajectory prediction calculated the independent movement of the target vehicle based on vehicle kinematic or dynamic modeling. Kalman filters are mainly used to track the vehicles' positions and predict their future states [7–9,25,26]. Trajectory prediction based on simple physical models and Kalman filters has the disadvantage of only being effective at predicting future states for a short time. Gaussian mixture modeling [10] or Monte Carlo path planning [27] is used to solve the short-term prediction problem. Efforts have been made to improve prediction performance by classifying the future behavior of vehicles before making physical state predictions. Finite behavior types are classified using behavior classification models, such as Bayesian networks [28], support vector machines [29,30], and HMM [31–33], and trajectories or risks corresponding to each behavior are predicted.

Interaction aware prediction: Interaction aware prediction methods consider interactions between vehicles. A relatively small number of such works have been published. Optimal predictions regarding future motions have been made using heuristic cost functions [34–36], data-driven random forest classifiers [37], and Markov decision processes [38] based on relative information between vehicles. In [34], 10 types of maneuvers are classified using HMM, and an Interacting Multiple Model (IMM) is used to model the vehicles' motion. An energy minimization cost function is applied to the results of a maneuver recognition module and a trajectory prediction module for each vehicle, and this is used as a module for predicting interactions. The use of optimization models has a limitation in that these models are significantly affected by how well the cost function is designed. In another method, the results of classifying lateral maneuvers using a random decision forest model based on data obtained in an actual road environment are combined with a Gaussian mixture regression model's probabilistic prediction trajectories [37]. Approaches based on real road data have the burden of needing to construct a large-scale dataset, and they may be overfitted for certain situations if data from a variety of driving environments cannot be obtained.

Recurrent networks-based prediction: Mozaffari et al. published a detailed investigation into deep learning-based vehicle behavior prediction [39]. Convolutional Neural Networks (CNNs) are used to predict surrounding vehicles' driving intentions and trajectories based on sensor data [40–44]. As CNN-based prediction methods lack a mechanism for reflecting time-series information, researchers have presented works that use RNNs and CNNs in combination to integrate the advantages of each model [19,45–47]. Researchers have also proposed a method that uses occupancy grid maps and an LSTM encoder–decoder model to probabilistically predict trajectories [18]. In [19,20], interactions with surrounding vehicles are modeled by a social pooling mechanism. The context vectors of trajectory and maneuver encoders are combined to predict the maneuver-specific future distributions of vehicles' positions [48]. The LSTM RNN model can be used for intelligent traffic management and route guidance by predicting traffic flow from a macro perspective [49] as well as microscopic vehicle movement trajectory prediction. A traffic control system that considers current and future traffic congestion conditions has improved a city's traffic flow [50]. Classification is performed on behaviors such as lane changes

and deceleration, which can be recognized by turn signals and brake lights in real road situations. In Neural Machine Translation (NMT), which is an important field that uses the sequence-to-sequence model, researchers have published results in which translation performance and learning efficiency are greatly improved by using an attention mechanism that selectively focuses on parts of the source text [51,52]. In the field of vehicle trajectory prediction, researchers have also begun to use attention mechanisms to improve prediction performance by focusing on information such as certain vehicles or time points [21,53–55]; however, additional research is still required on various methodologies that can effectively emulate the methods of judgment used by actual drivers. In most related studies, the future vehicle trajectory prediction models are trained and verified based on the Next Generation Simulation (NGSIM) [56,57] or HighD [23] datasets. Nevertheless, these studies have not sufficiently verified their performance on a variety of road geometries. Yoon et al. predicted realistic driving intentions by extracting road geometry data from detailed roadmap data and then using the extracted data as constraints in a prediction model [58].

3. Proposed Vehicle Trajectory Prediction Method

The vehicle trajectory prediction model was designed based on dynamics data such as the relative positions and speeds of the surrounding vehicles converted to the reference frame of the target vehicle. First, a basic LSTM encoder–decoder model was designed, and an attention mechanism was applied to increase the ability to understand the driving context of the encoder.

3.1. Problem Formulation

Our task was to predict the future trajectories of surrounding vehicles in various driving environments; this has an important effect on the performance of partially or fully automated vehicles. This study utilized the idea of a deep learning model, which has caused performance improvements compared to past efforts in fields such as NLP and NMT; these have the common point of predicting continuous future data based on a series of past data. For the performance of a prediction algorithm, providing high-quality, large-scale data is as important as the structure of a deep learning model. Fortunately, we could utilize public datasets containing the driving information of actual traffic vehicles [23,55,56]. The deep learning model's learning efficiency and prediction accuracy are improved by organizing the data in the dataset, which contains vast amounts of information, to prioritize the information that mainly affects the decisions of actual drivers. Another challenge is to acquire versatility such that the prediction ability, which is limited to the learning dataset's driving environment, can be used in a variety of situations. We addressed this challenge by noting that human drivers would consider vehicles driving along a predetermined road geometry similarly to vehicles driving straight.

3.2. Surrounding Vehicle Data Processing

We converted the data for surrounding vehicles based on a local coordinate system that was fixed on the center of the back face of the target vehicle, as illustrated in Figure 2 (Figure 2 shows some of the driving data used for learning). When human drivers or autonomous vehicles perceive a driving situation and interact with surrounding vehicles, they think from the perspective of a local coordinate system with their own field of view as the reference point. If global coordinates expressed the positions of vehicles, the data described by the numbers would be completely different to the target vehicle's viewpoint, even in the same situation. This could cause inefficiencies in learning and harm prediction performance.

In Figure 2, the green vehicle's heading direction is the x -axis and the direction perpendicular to that is the y -axis. The blue vehicles are the closest cars in each lane adjacent to the target vehicle, and they are the objects that must be examined with the greatest caution when considering a lane change. The traffic flow of each lane is expressed by the distance between the front and rear vehicles and their amount of change, as well as the location and speed of the ego (green) vehicle and the blue vehicles.

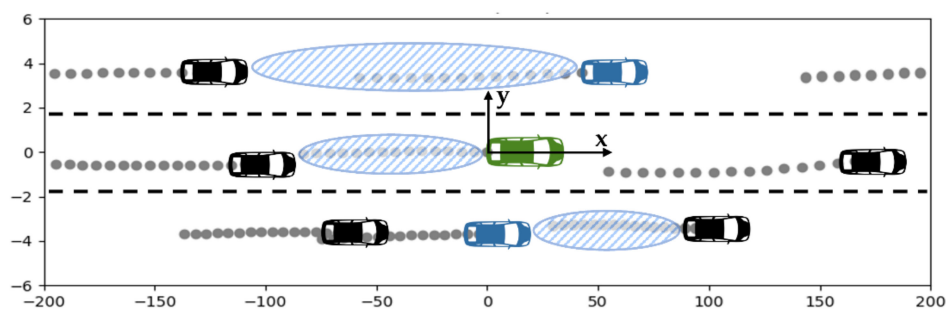


Figure 2. Coordinate system used for trajectory prediction. Green: target vehicle, blue: nearest vehicle in adjacent lane, patterned ellipse: front and rear inter-vehicle distance.

Tijerina et al. observed that lane changing durations last an average of 5.0 s on city streets and 5.8 s on highways [59]. Toledo et al. analyzed the duration of lane changes according to variables such as vehicle type, driving velocity, and relative distance. [60]. Previous studies, including [61,62], have reported lane change durations of approximately 5 s on average; therefore, we predicted the position of the target vehicles up to 5 s into the future based on information on the target vehicles and surrounding vehicles from the past 3 s.

3.3. Base LSTM Encoder–Decoder Trajectory Prediction Model

LSTM encoder–decoder models have been proposed in the field of machine translation [14,63]. To verify our proposed model’s performance and the effectiveness of the learning data configuration, the peaky LSTM encoder–decoder model illustrated in Figure 3 was created by referencing architecture [14] that was proposed in the field of machine translation. Decent prediction performance can be expected, even from a basic model, if the model uses large amounts of data that properly depict the situations to be learned [64,65].

- Input and output data: Our model’s input comprised the data related to the traffic flow of the target vehicle’s lane as well as the lanes to the left and right of the target vehicle for a fixed amount of time.

$$X = [x^{t-t_h}, x^{t-t_h+1}, \dots, x^{t-1}, x^t] \tag{1}$$

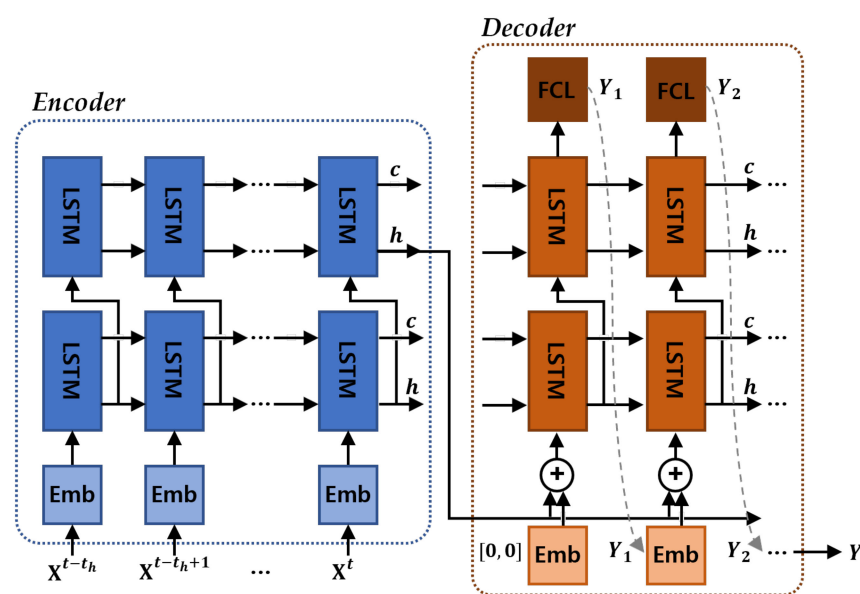


Figure 3. Peaky LSTM encoder–decoder architecture.

Here,

$$\mathbf{x}^t = [\mathbf{x}_c^t, \mathbf{x}_l^t, \mathbf{x}_r^t], \quad (2)$$

$$\mathbf{x}_c^t = [x_m^t, y_m^t, v_{x,m}^t, v_{y,m}^t, x_f^t, y_f^t, d_{mf}^t, \Delta v_{mf}^t, x_r^t, y_r^t, d_{mr}^t, \Delta v_{mr}^t] \quad (3)$$

Data from the current time t until the time before t_h were adopted as the input and data were sampled from up to 3 s before, at 5 Hz. The value \mathbf{x}^t at time t is the center, left, and right lane data (c: center, l: left, r: right), and each lane's data included the center vehicle's x - and y -axis position, velocity, distance to the vehicles in front and behind, and relative velocity (m: middle, f: front, r: rear). The center vehicles of each lane were the green or blue vehicles illustrated in Figure 2.

The model's output was the future coordinate of the target vehicle after time t_f .

$$\mathbf{Y} = [\mathbf{y}^{t+1}, \mathbf{y}^{t+2}, \dots, \mathbf{y}^{t+t_f}] \quad (4)$$

Here,

$$\mathbf{y}^t = [x_{tgt}^t, y_{tgt}^t] \quad (5)$$

- Encoder: The encoding layer receives data from each time as the input and sends it through the embedding and LSTM layers to convert it to hidden state vectors. The cell and hidden state vectors that are calculated in the LSTM for each time step are sent to the next step. The topmost LSTM layer's hidden state at the final time point acts as the context vector in which the driving information of the vehicles for a fixed amount of time is encoded. The LSTM has memory cells that summarize the past input sequences and store them, and these cells consist of the following gating mechanisms that properly combine the new input and memory information (Figure 4).

$$forget, \mathbf{f} = \sigma(\mathbf{x}^t \mathbf{W}_{xf} + \mathbf{h}^{t-1} \mathbf{W}_{hf} + \mathbf{b}_f) \quad (6)$$

$$input, \mathbf{i} = \sigma(\mathbf{x}^t \mathbf{W}_{xi} + \mathbf{h}^{t-1} \mathbf{W}_{hi} + \mathbf{b}_i) \quad (7)$$

$$update, \mathbf{g} = \tanh(\mathbf{x}^t \mathbf{W}_{xg} + \mathbf{h}^{t-1} \mathbf{W}_{hg} + \mathbf{b}_g) \quad (8)$$

$$output, \mathbf{o} = \sigma(\mathbf{x}^t \mathbf{W}_{xo} + \mathbf{h}^{t-1} \mathbf{W}_{ho} + \mathbf{b}_o) \quad (9)$$

$$cell\ state, \mathbf{c}^t = \mathbf{f} \odot \mathbf{c}^{t-1} + \mathbf{g} \odot \mathbf{i} \quad (10)$$

$$hidden\ state, \mathbf{h}^t = \mathbf{o} \odot \tanh(\mathbf{c}^t) \quad (11)$$

where $\sigma(x)$ is an activation function, \mathbf{W}_x and \mathbf{W}_h represent weight matrix for input and hidden state, \mathbf{b}_f is a bias vector of forget gate, and \mathbf{c}^t and \mathbf{h}^t denote cell and hidden state vectors at time step t .

- Decoder: The context vectors that summarize the past driving information are sent from the encoding layer to the decoder layer's input. The hidden states that are calculated in the LSTM layer for each time step are converted to x and y coordinates by the fully connected neural network layer. The position vectors that are ultimately produced as an output become the input of the next step, and a similar process is repeated until the goal prediction time is reached to determine a continuous future prediction position at each time point. The peaky LSTM encoder–decoder model connects the position vectors that become the input of each time step and the context vectors that are produced as an output by the encoding layer. The prediction performance can be improved by not sending the context vector solely on the decoder's first step and then using the past driving information at each step.
- Loss function: In deep learning models, training progresses in the direction of reducing the loss function. Our ultimate training goal was to determine prediction points at the closest distance to the actual future position. Therefore, we adopted the Root

Mean Square Error (RMSE), which corresponds to distance error, as the loss function. Additionally, more importance was placed on lateral accuracy than longitudinal accuracy [21].

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n \sum_{t=1}^{t_f} \{(\hat{x} - x)^2 + 2 \cdot (\hat{y} - y)^2\}} \quad (12)$$

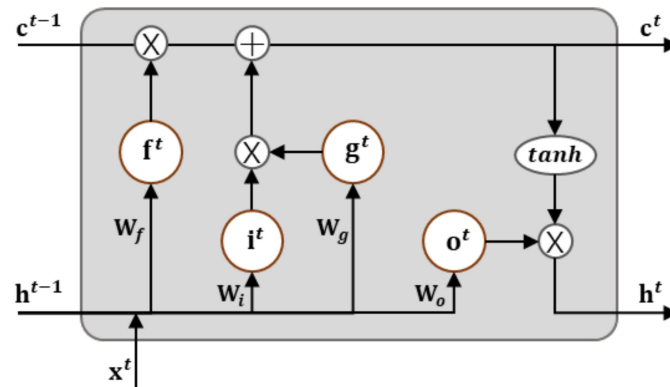


Figure 4. The structure of an LSTM cell.

Here, x , y are the true values at each time step, and \hat{x} , \hat{y} are the predicted positions.

3.4. Lane Stream Attention-Based LSTM Encoder–Decoder Trajectory Prediction Model

In the NMT field, the input information that is summarized as context vectors of fixed length is considered to cause an obstruction in improving the performance of the encoder–decoder architecture. Performance is improved by introducing a mechanism that can focus on the most relevant information in the context vector at each prediction time point [51,52]. In the challenge of vehicle trajectory prediction, unlike that of translation, it is effective to focus on information such as traffic lanes' driving flow rather than certain time points in the input information [21].

Our proposed lane stream attention-based LSTM encoder–decoder model was created by adding an attention mechanism to the basic model that separates and encodes the driving flow of each lane, as well as the target vehicle information, and determines its degree of importance at each time step, as described in Section 3.3 (Figure 5).

- Input and output data: The proposed attention-based model uses three encoders to depict the driving flow of the lanes that are adjacent to the target vehicle and one encoder that focuses on the target vehicle. The input data comprise central information that is considered with great caution when a human driver adjusts the vehicle's velocity or changes lanes [66,67]. The lane driving flow encoder's input is

$$\mathbf{x}_{\text{lane}}^t = \left[x_m^t, y_m^t, v_{x,m}^t, v_{y,m}^t, x_f^t, y_f^t, d_{mf}^t, \Delta v_{mf}^t, x_r^t, y_r^t, d_{mr}^t, \Delta v_{mr}^t \right] \quad (13)$$

and the target vehicle encoder's input is

$$\mathbf{x}_{\text{tgt}}^t = \left[x_{\text{tgt}}^t, y_{\text{tgt}}^t, v_{x,\text{tgt}}^t, v_{y,\text{tgt}}^t, x_{\text{front}}^t, x_{\text{rear}}^t, x_{\text{left}}^t, x_{\text{right}}^t, s_{\text{turn}}, s_{\text{brake}} \right]. \quad (14)$$

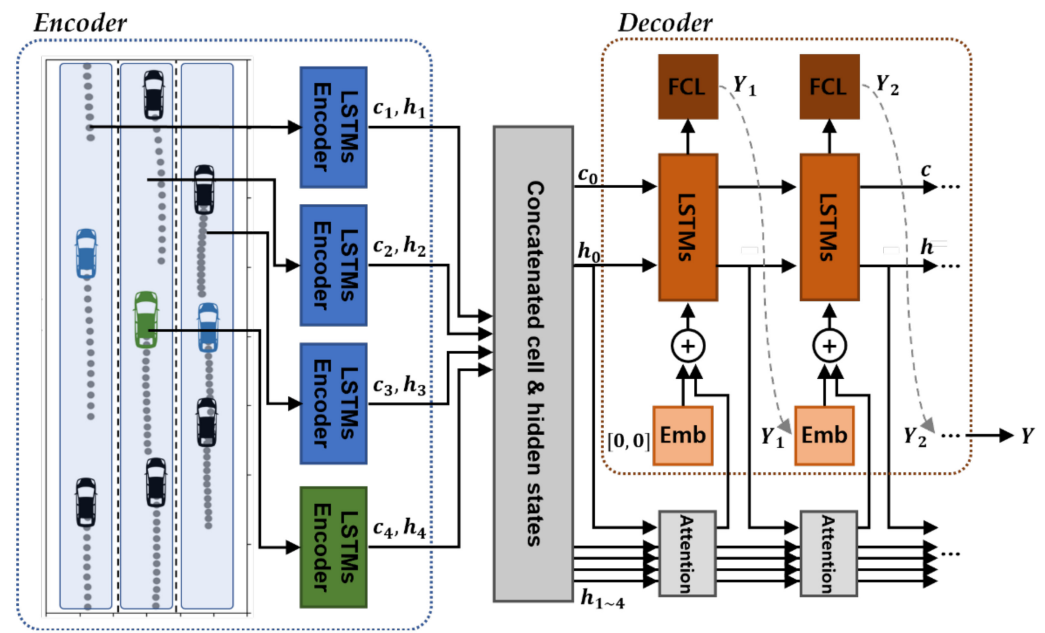


Figure 5. Lane stream attention-based LSTM encoder–decoder architecture.

The lane information input at time t is similar to those of the peaky LSTM model's lane input data. The target vehicle information input includes the x - and y -axis positions, velocity, and longitudinal positions of the surrounding vehicles in the front, back, left, and right directions. In addition, the states of the turn signals and brake lights are added as they provide the most important hints when an actual human driver predicts a surrounding vehicle's movement. As brake lights generally come on unconditionally when the brake pedal is pressed, they are set to the "on" state when decelerating by more than a certain velocity change. An average of 52–75% of actual drivers use turn signals in situations such as turning at intersections or changing lanes [68,69]. We set the turn signals to the "on" state during 60% of lane-change sequences in the training data. The model's output is the target vehicle's future position coordinates after time t_f .

- Attention: The final hidden state that is produced as an output by each encoder summarizes the sequence of driving information, and the influence of the most recent information is strongly reflected; therefore, it is appropriate to use this as a context vector that predicts the future position. The weight of each context vector is calculated as follows:

$$\text{hidden states, } \mathbf{h}_0 = \text{concat}(\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3, \mathbf{h}_4) \quad (15)$$

$$\mathbf{a}_n^t = \text{FCNN}(\text{concat}(\mathbf{h}^t, \mathbf{h}_n)), n = 1, 2, 3, 4 \quad (16)$$

$$\text{attention weights, } \mathbf{A}^t = \text{softmax}(\text{concat}(\mathbf{a}_1^t, \mathbf{a}_2^t, \mathbf{a}_3^t, \mathbf{a}_4^t)) \quad (17)$$

The context vector \mathbf{h}^t at prediction time t is a hidden state vector that includes the past information and the predicted position in the LSTM layer, and the initially inputted \mathbf{h}_0 is connected to the final hidden states that are produced as outputs by each encoder. A neural network was utilized to quantify the importance of the encoding layer's output \mathbf{h}_{1-4} , and a softmax function was used to normalize the values to between 0 and 1. The hidden state vectors were multiplied by the weights and added to the embedded input to become the next step's LSTM input value. The method for calculating the attention weights was developed by referencing models proposed by previous studies on machine translation and trajectory prediction [21,52].

- Encoder–decoder: The encoding and decoding layers are similar to those described in the basic LSTM structure. However, in an attention-based model, a total of four

encoding layers are utilized and a context vector that reflects the attention weights is added to the decoding layer. To prevent the challenge of the model becoming overfitted to the training data in the training process, the data are scaled and dropout is applied to the LSTM layer.

4. Evaluation of Vehicle Trajectory Prediction Model with Traffic Dataset

This section presents the training results of the proposed trajectory prediction model using a natural traffic dataset. The data required for learning were extracted from publicly available datasets, and additionally necessary values were subjected to a pre-processing process. For proper performance evaluation, prediction results of different models using the same dataset were compared.

4.1. HighD Traffic Dataset

We used the HighD dataset, which includes publicly available traffic data, to train and evaluate the proposed model [23]. The HighD dataset is a large-scale, naturalistic vehicle trajectory dataset that includes driving data from more than 110,000 cars and trucks captured by drones on highways in Germany (Figure 6). A total of 16.4 h of data were captured from six different road sections, and the length of each section was approximately 420 m. The videos were recorded at 25 fps in 4K resolution, and a computer vision algorithm was used to automatically extract the vehicles' data. The extracted information on each vehicle included the size, type, driving direction, position, velocity, acceleration, number of lane changes, and IDs of surrounding vehicles. The advantage of the HighD dataset is that positions are measured with an error of less than 10 cm from a vast amount of high-quality raw data. Additionally, various types of information, such as the surrounding vehicles' information and lane changes, have been preprocessed. As it already includes various information that describes driving situations, it is very efficient for configuring data for learning.

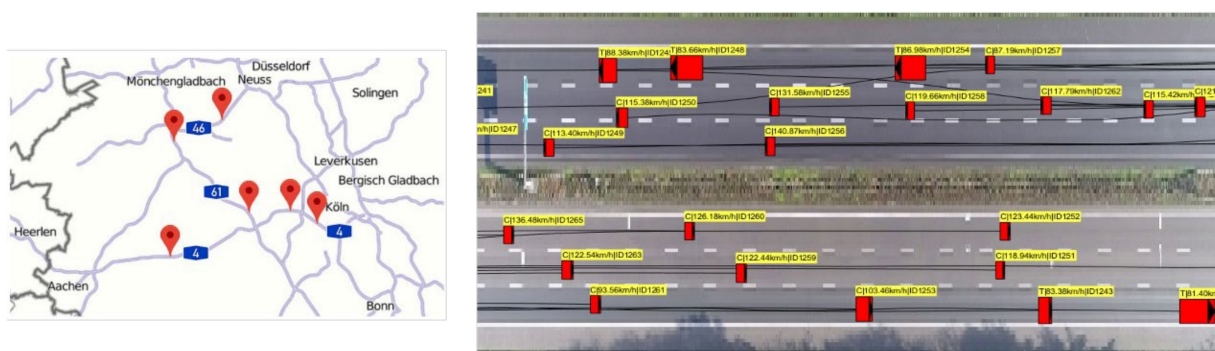


Figure 6. HighD dataset: highway traffic dataset with drone [23].

We extracted the input data for each vehicle that was required for training and added information such as turn signals. The target vehicle and surrounding vehicles' position and velocity data were all converted to the standard local coordinate system illustrated in Figure 2. If there was an empty position among eight vehicles around the target, the relative position and speed could not be calculated, so it was replaced with virtual data from 300 m that did not affect driving. To reduce unnecessary burden on the prediction model, 25 fps data were down-sampled to 5 fps, and the data that were divided into evenly spaced intervals were used to obtain lane-changing sequences. We divided the preprocessed dataset into training (70%), validation (10%), and testing (20%) sets, and then performed the training.

4.2. Training Results of Trajectory Prediction Model

The proposed model predicts future positions at up to 5 s into the future at 5 Hz. We compared the actual future positions and prediction results at 1 s intervals using the RMSE metric to verify the model's prediction performance. Table 1 compares the results of several baseline models that were published using a similar dataset. The baseline models were divided into groups comprising models that use only the position data of the vehicles and models that also use additional information such as velocity and maneuvers. The models that learn using solely position data include Convolutional Social (CS)-LSTM (a social encoder–decoder using convolutional pooling [19]), Non Local Social (NLS)-LSTM (which combines local and non-local operations for social pooling [20]), and Multi Head Attention (MHA)-LSTM (which uses multi-head dot product attention [54]). The models that also use additional information such as vehicle velocity, acceleration, and class include MHA-LSTM (MHA-LSTM with additional features [54]), Encoder Decoder (ED)-LSTM (a basic LSTM encoder–decoder model), P-LSTM (peaky LSTM encoder–decoder model), ED-LSTM with CS (which uses the convolutional pooling concept [19]), and Lane Stream (LS)-LSTM (the proposed model in this study, which uses lane stream attention).

Table 1. RMSEs in meters over a 5 s prediction horizon for the proposed and baseline models.

Prediction Horizon (s)	Position-Based Methods			Position + Other Features-Based Methods				
	CS-LSTM	NLS-LSTM	MHA-LSTM	MHA-LSTM(+f)	ED-LSTM	P-LSTM	ED-LSTM +CS	LS-LSTM
1	0.22	0.20	0.19	0.06	0.30	0.30	0.32	0.30
2	0.61	0.57	0.55	0.09	0.50	0.43	0.61	0.38
3	1.24	1.14	1.10	0.24	0.76	0.60	0.98	0.45
4	2.10	1.90	1.84	0.54	1.08	0.88	1.45	0.60
5	3.27	2.91	2.78	1.18	1.48	1.26	1.99	0.88

As presented in Tables 1 and 2, the models that learn solely from position information show relatively low performance, and it is effective to use additional information such as velocity to accurately predict future positions. This is because moment-to-moment relative positions between vehicles, as well as each vehicle's dynamics, have an important effect. If the surrounding vehicles' data are converted to a local frame that is fixed on the target vehicle, as was performed in our proposed method, excellent prediction results can be expected from a P-LSTM model that inputs the context vectors that have been converted by a basic encoder–decoder model at each time step. The proposed LS-LSTM model's long-term prediction performance was excellent compared to the other baseline models. This approach uses a mechanism of primarily processing and inputting the driving situation in terms of the surrounding lanes' driving flow and the target vehicle's information and then learning the importance of each item of information; it can be observed that this mechanism was effective.

Table 2. Longitudinal and lateral RMSEs in meters.

Prediction Horizon (s)	Longitudinal Position				Lateral Position			
	ED-LSTM	P-LSTM	ED-LSTM +CS	LS-LSTM	ED-LSTM	P-LSTM	ED-LSTM +CS	LS-LSTM
1	0.25	0.28	0.31	0.28	0.18	0.08	0.09	0.09
2	0.39	0.40	0.58	0.36	0.31	0.15	0.17	0.13
3	0.62	0.56	0.95	0.41	0.44	0.22	0.26	0.18
4	0.92	0.83	1.41	0.54	0.56	0.30	0.36	0.25
5	1.32	1.19	1.94	0.81	0.67	0.39	0.46	0.33

We compared the models' learning curves to verify the learning effectiveness of the proposed attention mechanism (Figure 7). When the attention mechanism is applied, the learning speed increases and, ultimately, a low test loss is reached in a stable manner that is clearly distinct from the other models. The peaky mechanism inputs the encoder's context vectors in the decoder at each moment and, when it is not used, the loss becomes sufficiently low for the training dataset. However, the accuracy was poor for the test data.

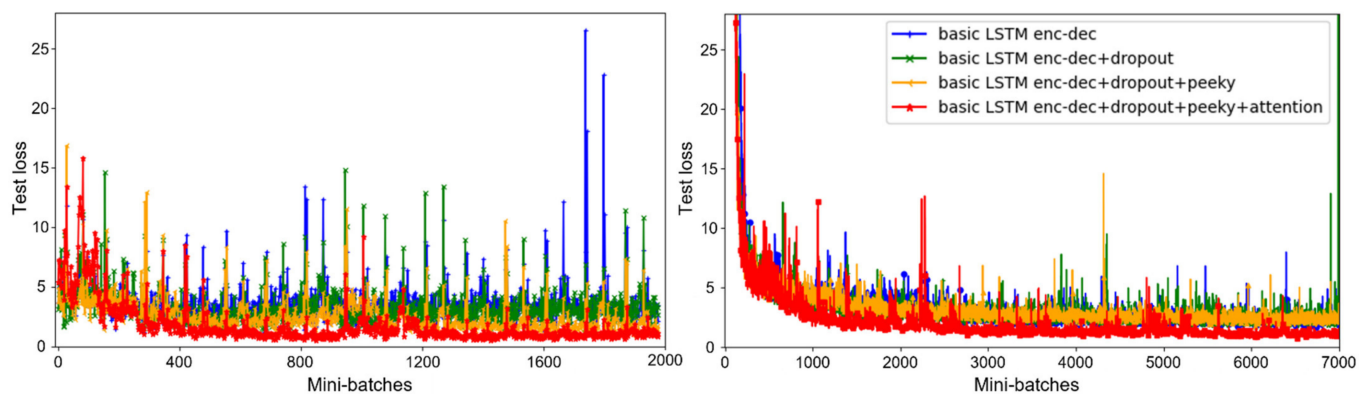


Figure 7. Learning curves: test and training loss.

Figure 8 illustrates the lane change and lane-keeping sequences when using the trained model on the test dataset. In the lane change sequence in Figure 8a, the LS-LSTM model predicted the lane change more quickly than the P-LSMT model and reached the actual future trajectory. The attention mechanism calculates the weights for objects that must receive focus in real-time according to the surrounding vehicles' information and the target vehicle's data. In the lane-keeping sequence in Figure 8b, it can be observed that the center lane and the target vehicle's weight values were relatively high.

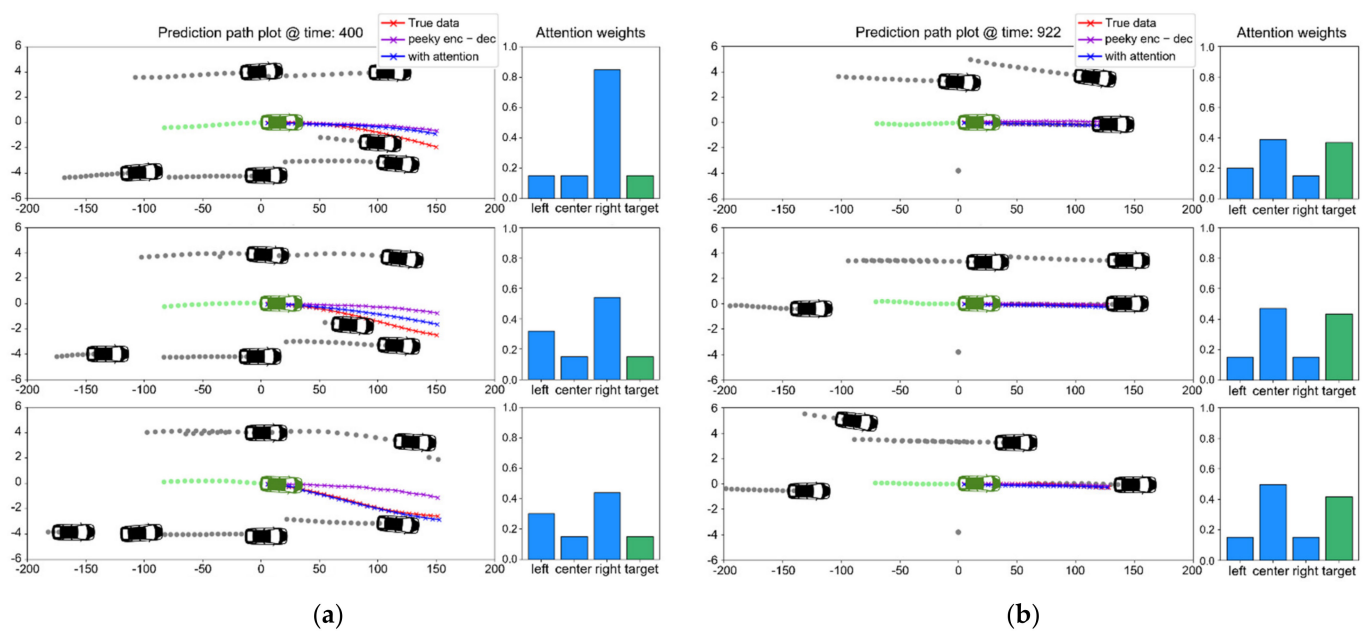


Figure 8. Examples of trajectory prediction: (a) lane changing and (b) lane keeping.

5. Road Geometry Linearization Method for Trajectory Prediction in Real Driving Environments

In order to expand the scope of the dataset limited to straight roads, a simulation environment was built based on actual road map data. A traffic event generation model that changes each vehicle's longitudinal and lateral behavior over time was applied to collect traffic data in various situations in a virtual urban environment. This section proposes a method to simplify complex driving situations by linearizing a curved road's reference path, as shown in Figure 9.

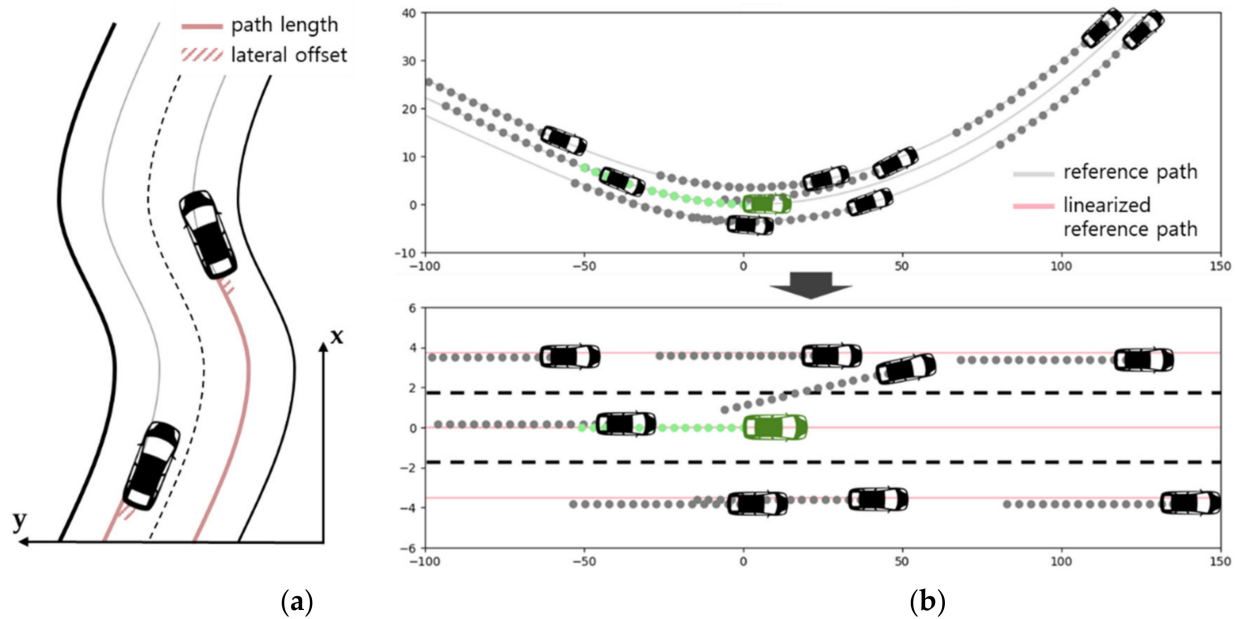


Figure 9. Road geometry linearization method: (a) reference of frame and (b) path linearization.

5.1. Road Geometry Linearization Method

The future trajectory prediction model, which was developed using a highway traffic driving dataset, exhibited excellent performance. However, most actual driving is undertaken on roads with a variety of geometries rather than completely straight roads. Therefore, to apply the trajectory prediction model to actual driving situations, one of two methods is necessary. The first is to collect driving data from roads with sufficiently different geometries and have the deep learning model consider geometry when learning. The second is to simplify the driving environment so that it is similar to a straight road scenario [70]. We used a method that linearized the road geometry, as illustrated in Figure 9, so that the proposed trajectory prediction model could function in a variety of environments.

When human drivers drive on a road such as the one illustrated in Figure 9a, the driver of the vehicle on the left side does not think that the vehicle on the right side is changing lanes to the left but rather thinks that it is moving straight along a regularly shaped road. This means that when a surrounding vehicle's intentions are judged, how the vehicle is moving longitudinally and laterally in reference to each lane's center is more important than the vehicle's absolute position or heading direction. Therefore, we linearized the driving situations as illustrated in Figure 9b by using the surrounding vehicles' progress distance and lateral offset regarding a reference trajectory corresponding to the center line of each lane. The surrounding vehicles were rotationally transformed to the local standard coordinate system fixed on the center of the back face of the target vehicle, as illustrated in Figure 2.

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos(\psi) & -\sin(\psi) \\ \sin(\psi) & \cos(\psi) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (18)$$

Here, ψ is the target vehicle's yaw angle. The linearized position's x value is the longitudinal progress distance based on the point at which the rotationally transformed surrounding vehicle's driving lane reference trajectory crosses the y -axis. The linearized position's y value is calculated by adding the spacing between lanes and the distance of lateral divergence from the reference trajectory.

5.2. Complex Traffic Driving Data Generation in Simulation Environment

The IPG CarMaker 10.1 virtual test-driving environment was used to acquire complex driving data, including curves and left/right turning sections, to verify the trajectory pre-

dition model. This commercial software provides features that can realistically implement static environments, such as road models, buildings, and traffic signals, as well as dynamic driving scenarios such as traffic vehicles and pedestrians. IPG CarMaker is used for vehicle design and verification by car manufacturers or for autonomous driving algorithm development by research institutes. We configured the traffic vehicles in a simulation environment based on actual road data provided by IPG Automotive Korea and acquired the driving data. The Sangam autonomous vehicle test-driving district in Seoul, South Korea, was modeled in a virtual environment, as illustrated in Figure 10a. Figure 10b illustrates a scene from the simulation used to acquire the data.



Figure 10. Virtual test-driving environment based on real road data: (a) simulation environment and (b) virtual driving scene.

To create various driving scenarios with the traffic vehicles, we designed a traffic maneuver generation model (Figure 11). Each of the traffic vehicles were assigned normal, longitudinal, and lateral events at fixed ranges of time intervals. During normal events, the vehicles drive along the selected trajectory at a fixed velocity. When longitudinal or lateral events occur, the vehicles change velocity or move laterally. The traffic maneuver generation model controls the probability of each event occurring as well as the amount of acceleration/deceleration and the lateral movement distance and velocity.

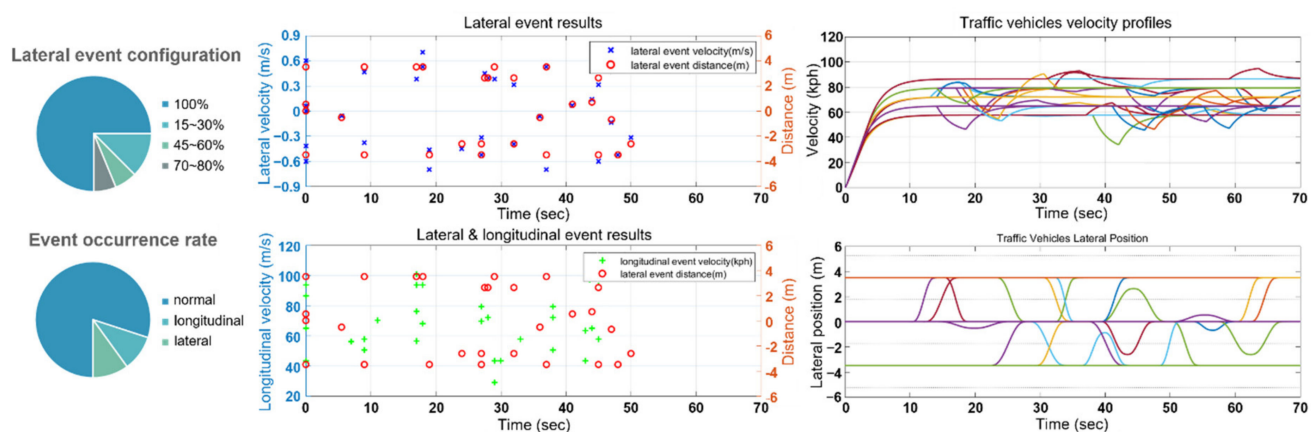


Figure 11. Traffic event generation results.

The variables determined by lateral events are the lateral offset and duration. When the lateral offset is smaller than the lane width, the vehicles do not change lanes completely but move to the left or right and then return to the existing lane to model actual vehicles moving in reference to the lane center.

6. Experimental Evaluation

This section presents the evaluation of the proposed road geometry linearization method and trajectory prediction model based on the traffic scenarios of the virtual driving simulation. The same trajectory prediction model was applied to compare whether the route straightening model could convert curved driving data similarly to the HighD dataset driving environment.

6.1. Evaluation of Trajectory Prediction Model with Path Linearization Method

By using the traffic maneuver generation model, the ratio at which the vehicles change lanes can be controlled. Datasets that corresponded to the following three scenarios in the Sangam Digital Media City (DMC) virtual driving environment were created:

- Scenario 1: Driving on a straight road section that is approximately 1.2 km long with lane changes (a driving environment like that in the HighD dataset).
- Scenario 2: Repeatedly driving on a complex, closed-loop road section that is approximately 1.5 km long and has straight road and curved road sections and intersections (left and right turns) without lane changes.
- Scenario 3: Driving on a similar road to the one described in Scenario 2 with lane changes.

Table 3 presents the results of applying the model that was trained using the HighD dataset after preprocessing the acquired data using the method in Section 3.

Table 3. Longitudinal and lateral RMSEs in meters with simulation scenarios.

Prediction Horizon (s)	Scenario 1		Scenario 2		Scenario 3	
	P-LSTM (Long, Lat)	LS-LSTM (Long, Lat)	LS-LSTM w/o Linearization	LS-LSTM w/ Linearization	LS-LSTM w/o Linearization	LS-LSTM w/ Linearization
1	0.35, 0.09	0.37, 0.07	0.84, 0.39	0.80, 0.06	0.67, 0.48	0.73, 0.07
2	0.60, 0.15	0.68, 0.12	1.21, 1.47	1.02, 0.10	1.10, 1.75	1.14, 0.13
3	0.80, 0.20	0.75, 0.16	1.65, 3.12	1.37, 0.11	1.45, 3.69	1.54, 0.16
4	1.12, 0.24	0.83, 0.18	2.28, 5.28	1.54, 0.13	2.17, 6.26	1.83, 0.19
5	1.50, 0.28	0.91, 0.20	3.56, 7.87	1.96, 0.13	3.35, 9.37	2.31, 0.22

The results of Scenario 1, a scenario similar to a highway driving environment, are similar to the training results in Table 2. The longitudinal prediction performance was reduced because the acceleration and deceleration regions caused by the longitudinal events assigned to the simulation vehicles were more rapid than the actual highway driving data. However, the lateral prediction performance was better because of the continuous and accurate position and velocity data in the simulation. The results of Scenario 1 indicate that it is important to use a variety of velocity and acceleration data in the training. Additionally, it is necessary to support an algorithm that precisely recognizes and tracks vehicles.

In Scenarios 2 and 3, in which vehicles drive on complex roads that include curves and intersections, when the linearization method is not used, a significant error occurs to the extent that the future trajectory cannot be meaningfully predicted. By using the trajectory linearization method, the lateral prediction results were improved so that they were similar to those of the straight road sections, as illustrated in Figure 12. On the complex roads' curves and intersections, a fairly large error occurred in the longitudinal direction despite trajectory linearization because the vehicles' velocities changed with relatively large acceleration.

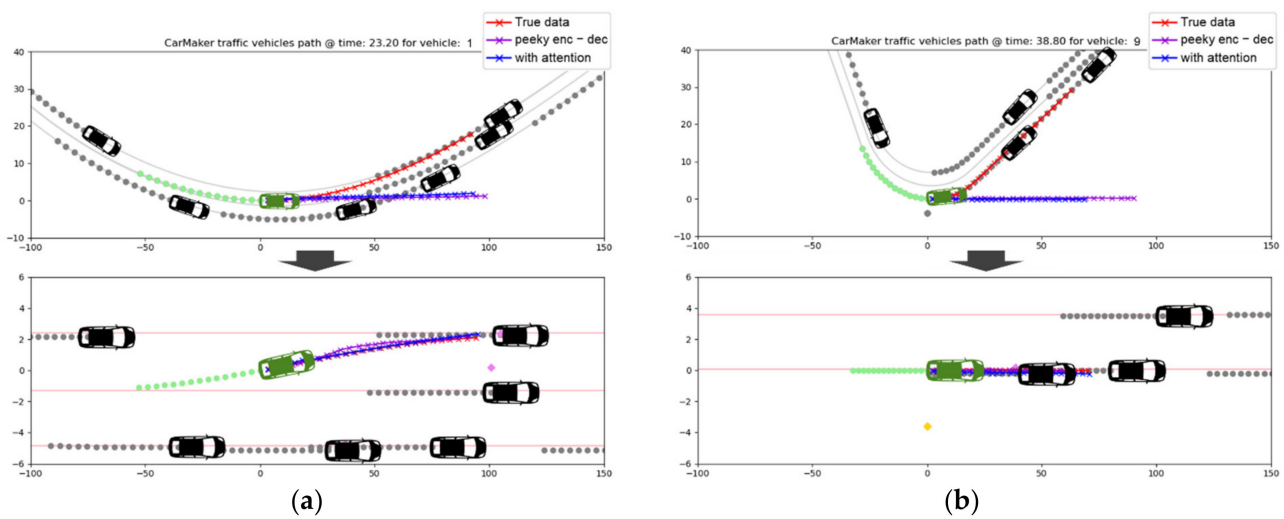


Figure 12. Trajectory prediction with simulation data: (a) lane changing and (b) lane keeping.

6.2. Discussion

By using the traffic maneuver generation model, the ratio at which the vehicles change lanes could be controlled. Datasets that corresponded to the following three scenarios in the Sangam DMC virtual driving environment were created. This work performed long-term trajectory prediction of surrounding vehicles in a general road environment to improve the capabilities of autonomous vehicles. The framework of the proposed trajectory prediction method was configured as shown in Figure 1. The results of training and testing the trajectory prediction model using the HighD dataset are presented in Tables 1 and 2. The long-term trajectory prediction performance was stably improved by applying the attention mechanism to the relative information of the target vehicle and surrounding vehicles in adjacent lanes. A road geometry linearization method was introduced to develop existing studies that were limited to the dataset environment of straight highways. Table 3 presents the trajectory prediction results based on the path linearization mechanism in the complex driving simulation traffic data. The prediction error was reduced by 76.7 percent by applying the proposed path linearization method in the complex driving scenario (Scenario 3). The deep learning model trained with the publicly available dataset could be applied to various road environments by simplifying the complex driving environments with the proposed trajectory prediction algorithm.

As road geometry linearization was applied based on the original reference path, it is essential to correctly determine the reference driving path of surrounding vehicles. The reference path of vehicles driving along curved roads was simplified to each lane's middle line in this work. In order to accurately convert a complex natural driving environment into a straightened frame, a follow-up study is needed to find a reference route that vehicles generally travel on according to the curvature of the road and the driving speed or traffic direction. For example, most drivers may drive out-in-out for driving efficiency on certain roads, intentionally biasing one side to adjust the spacing with adjacent lanes, or the route itself may be complex, such as a merging section.

7. Conclusions

This study presented an LS attention-based LSTM encoder–decoder model and a road shape linearization method for predicting the future trajectory of surrounding vehicles. When changing lanes or adjusting speed, drivers consider relative information with surrounding vehicles; each of them is given importance. The proposed attention mechanism could implement the driver pattern by selectively focusing on adjacent lanes and the target vehicle to predict the future trajectories of vehicles. We verified the proposed model in terms of learning speed and prediction accuracy when using test data corresponding to 20%

of the entire dataset. The attention mechanism determined the focused object in real-time according to the driving situation and improved the long-term prediction performance. In addition, the road geometry linearization method was applied so that the learning model that was developed based on straight road data could be used in various driving environments. The linearization mechanism was implemented in the same way real drivers perceive they are going straight when driving along a curved road shape. Traffic driving scenarios were implemented in virtual test-drive environments based on real road data, and the trajectory prediction algorithm was verified. The performance of the proposed trajectory prediction model was verified by comparing it with models using the same dataset. The improved predictive performance of the path linearization method on curved roads was evaluated in a simulated complex traffic scenario. The importance of our proposed trajectory prediction method is summarized as follows:

- The proposed lane stream attention-based trajectory prediction model improved long-term prediction accuracy by 25.4% compared to other methods. The ability to transform the context of each driving situation of the attention mechanism applied to the encoder–decoder model can predict the long-term trajectory more accurately.
- The proposed road shape linearization method simplifies the complex real road situation and expands the application range of the trajectory prediction model. In the complex traffic scenario acquired in the virtual driving environment, the distance error of the trajectory prediction model with the path linearization method was reduced by 76.7% compared to the result without the method. It is a more efficient and realistic method that can be applied to autonomous vehicles that drive on real roads rather than building large-scale traffic datasets on roads of numerous shapes.

The proposed method evaluated the accuracy in a virtual driving environment, including a curved road and a HighD dataset. Although the road shape linearization method can simplify the curved road, the prediction distance error for vehicles traveling with velocities and accelerations outside the range of the model training data was increased. In particular, the prediction accuracy for a situation in which the speed was suddenly reduced or stopped while entering an intersection that was not in the training data was lowered. In the future, to apply the predictive model to urban driving or slow-moving situations, we plan to use a combination of datasets such as NGSIM in the lower speed range for training purposes.

Author Contributions: Conceptualization, D.Y.; software, D.Y. and H.L.; validation, D.Y.; formal analysis, D.Y. and T.K.; investigation, D.Y., H.L. and T.K.; data curation, D.Y.; writing—original draft preparation, D.Y.; writing—review and editing, D.Y., H.L. and T.K.; visualization, D.Y. and T.K.; supervision, S.-H.H.; funding acquisition, S.-H.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Ministry of Science and ICT, Korea, under the Information Technology Research Center support program (IITP-2021-2018-0-01426), supervised by the Institute for Information and Communications Technology Planning and Evaluation, and the Technology Innovation Program (20013794, Center for Composite Materials and Concurrent Design) funded by the Ministry of Trade, Industry and Energy.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Nilsson, J.; Silvin, J.; Brannstrom, M.; Coelingh, E.; Fredriksson, J. If, When, and How to Perform Lane Change Maneuvers on Highways. *IEEE Intell. Transp. Syst. Mag.* **2016**, *8*, 68–78. [[CrossRef](#)]
2. Sivaraman, S.; Trivedi, M.M. Dynamic Probabilistic Drivability Maps for Lane Change and Merge Driver Assistance. *IEEE Trans. Intell. Transp. Syst.* **2014**, *15*, 2063–2073. [[CrossRef](#)]
3. Zhang, Y.; Lin, Q.; Wang, J.; Verwer, S.; Dolan, J.M. Lane-change intention estimation for car-following control in autonomous driving. *IEEE Trans. Intell. Veh.* **2018**, *3*, 276–286. [[CrossRef](#)]
4. Vahidi, A.; Sciarretta, A. Energy saving potentials of connected and automated vehicles. *Transp. Res. Part C Emerg. Technol.* **2018**, *95*, 822–843. [[CrossRef](#)]

5. Lindner, L.; Sergiyenko, O.; Rivas-López, M.; Ivanov, M.; Rodríguez-Quiñonez, J.C.; Hernández-Balbuena, D.; Flores-Fuentes, W.; Tyrsa, V.; Muerrieta-Rico, F.N.; Mercorelli, P. Machine vision system errors for unmanned aerial vehicle navigation. In Proceedings of the 2017 IEEE 26th International Symposium on Industrial Electronics (ISIE), Edinburgh, UK, 19–21 June 2017; pp. 1615–1620.
6. Ivanov, M.; Sergiyenko, O.; Tyrsa, V.; Mercorelli, P.; Kartashov, V.; Hernandez, W.; Sheiko, S.; Kolendovska, M. Individual Scans Fusion in Virtual Knowledge Base for Navigation of Mobile Robotic Group with 3D TVS. In Proceedings of the IECON 2018—44th Annual Conference of the IEEE Industrial Electronics Society, Washington, DC, USA, 21–23 October 2018; pp. 3187–3192.
7. Ammoun, S.; Nashashibi, F. Real time trajectory prediction for collision risk estimation between vehicles. In Proceedings of the 2009 IEEE 5th International Conference on Intelligent Computer Communication and Processing, Cluj-Napoca, Romania, 27–29 August 2009; pp. 417–422.
8. Barrios, C.; Motai, Y. Improving Estimation of Vehicle's Trajectory Using the Latest Global Positioning System With Kalman Filtering. *IEEE Trans. Instrum. Meas.* **2011**, *60*, 3747–3755. [[CrossRef](#)]
9. Kim, B.; Yi, K. Probabilistic and Holistic Prediction of Vehicle States Using Sensor Fusion for Application to Integrated Vehicle Safety Systems. *IEEE Trans. Intell. Transp. Syst.* **2014**, *15*, 2178–2190. [[CrossRef](#)]
10. Wiest, J.; Hoffken, M.; Kresel, U.; Dietmayer, K. Probabilistic trajectory prediction with Gaussian mixture models. In Proceedings of the Intelligent Vehicles Symposium (IV), Madrid, Spain, 3–7 June 2012. [[CrossRef](#)]
11. Gindele, T.; Brechtel, S.; Dillmann, R. Learning Driver Behavior Models from Traffic Observations for Decision Making and Planning. *IEEE Intell. Transp. Syst. Mag.* **2015**, *7*, 69–79. [[CrossRef](#)]
12. Mikolov, T.; Karafiát, M.; Burget, L.; Cernocký, J.; Khudanpur, S. Recurrent neural network based language model. In Proceedings of the Interspeech 2010, 11th Annual Conference of the International Speech Communication Association, Chiba, Japan, 26–30 September 2010; pp. 1045–1048.
13. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)]
14. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv* **2014**, arXiv:1406.1078.
15. Graves, A. Generating Sequences With Recurrent Neural Networks. *arXiv* **2013**, arXiv:1308.0850.
16. Karpathy, A. The Unreasonable Effectiveness of Recurrent Neural Networks. 2016. Available online: <http://karpathy.github.io/2015/05/21/rnn-effectiveness> (accessed on 23 October 2021).
17. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to sequence learning with neural networks. In Proceedings of the 27th International Conference on Neural Information Processing Systems, Montréal, QC, Canada, 8–11 December 2014; pp. 3104–3112.
18. Park, S.H.; Kim, B.; Kang, C.M.; Chung, C.C.; Choi, J.W. Sequence-to-sequence prediction of vehicle trajectory via LSTM encoder-decoder architecture. In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 26–30 June 2018; pp. 1672–1678. [[CrossRef](#)]
19. Deo, N.; Trivedi, M.M. Convolutional Social Pooling for Vehicle Trajectory Prediction. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, UT, USA, 18–22 June 2018; pp. 1549–15498.
20. Messaoud, K.; Yahiaoui, I.; Verroust-Blondet, A.; Nashashibi, F. Non-local Social Pooling for Vehicle Trajectory Prediction. In Proceedings of the 2019 IEEE Intelligent Vehicles Symposium (IV), Paris, France, 9–12 June 2019; pp. 975–980.
21. Yan, J.; Peng, Z.; Yin, H.; Wang, J.; Wang, X.; Shen, Y.; Stechele, W.; Cremers, D. Trajectory prediction for intelligent vehicles using spatial-attention mechanism. *IET Intell. Transp. Syst.* **2020**, *14*, 1855–1863. [[CrossRef](#)]
22. Alahi, A.; Goel, K.; Ramanathan, V.; Robicquet, A.; Fei-Fei, L.; Savarese, S. Social LSTM: Human Trajectory Prediction in Crowded Spaces. In Proceedings of the 29th IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 961–971. [[CrossRef](#)]
23. Krajewski, R.; Bock, J.; Kloeker, L.; Eckstein, L. The highD Dataset: A Drone Dataset of Naturalistic Vehicle Trajectories on German Highways for Validation of Highly Automated Driving Systems. In Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, 15 April 2018; pp. 2118–2125.
24. Lefèvre, S.; Vasquez, D.; Laugier, C. A survey on motion prediction and risk assessment for intelligent vehicles. *ROBOMECH J.* **2014**, *1*, 1–14. [[CrossRef](#)]
25. Barth, A.; Franke, U. Where will the oncoming vehicle be the next second? In Proceedings of the 2008 IEEE Intelligent Vehicles Symposium, Eindhoven, The Netherlands, 4–6 June 2008; pp. 1068–1073.
26. Toledo-Moreo, R.; Zamora-Izquierdo, M.A. IMM-Based Lane-Change Prediction in Highways With Low-Cost GPS/INS. *IEEE Trans. Intell. Transp. Syst.* **2009**, *10*, 180–185. [[CrossRef](#)]
27. Broadhurst, A.; Baker, S.; Kanade, T. Monte Carlo road safety reasoning. In Proceedings of the IEEE Intelligent Vehicles Symposium, Las Vegas, NV, USA, 6–8 June 2005; pp. 319–324.
28. Schreier, M.; Willert, V.; Adamy, J. Bayesian, maneuver-based, long-term trajectory prediction and criticality assessment for driver assistance systems. In Proceedings of the 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), Qingdao, China, 8–11 October 2014; pp. 334–341.
29. Mandalia, H.M.; Salvucci, M.D.D. Using Support Vector Machines for Lane-Change Detection. *Proc. Hum. Factors Ergon. Soc. Annu. Meet.* **2005**, *49*, 1965–1969. [[CrossRef](#)]

30. Aoude, G.S.; Luders, B.D.; Lee, K.K.; Levine, D.S.; How, J.P. Threat assessment design for driver assistance system at intersections. In Proceedings of the 13th International IEEE Conference on Intelligent Transportation Systems, Funchal, Portugal, 19–22 September 2010; pp. 1855–1862.
31. Laugier, C.; Paromtchik, I.E.; Perrollaz, M.; Yong, M.Y.; Yoder, J.-D.; Tay, C.; Mekhnacha, K.; Nègre, A. Probabilistic Analysis of Dynamic Scenes and Collision Risks Assessment to Improve Driving Safety. *IEEE Intell. Transp. Syst. Mag.* **2011**, *3*, 4–19. [[CrossRef](#)]
32. Liu, P.; Kurt, A.; Ozguner, U. Trajectory prediction of a lane changing vehicle based on driver behavior estimation and classification. In Proceedings of the 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), Qingdao, China, 8–11 October 2014; pp. 942–947.
33. Schlechtriemen, J.; Wedel, A.; Hillenbrand, J.; Breuel, G.; Kuhnert, K.-D. A lane change detection approach using feature ranking with maximized predictive power. In Proceedings of the 2014 IEEE Intelligent Vehicles Symposium Proceedings, Dearborn, MI, USA, 8–11 June 2014; pp. 108–114.
34. Deo, N.; Rangesh, A.; Trivedi, M.M. How Would Surround Vehicles Move? A Unified Framework for Maneuver Classification and Motion Prediction. *IEEE Trans. Intell. Veh.* **2018**, *3*, 129–140. [[CrossRef](#)]
35. Bahram, M.; Hubmann, C.; Lawitzky, A.; Aeberhard, M.; Wollherr, D. A Combined Model- and Learning-Based Framework for Interaction-Aware Maneuver Prediction. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 1538–1550. [[CrossRef](#)]
36. Lawitzky, A.; Althoff, D.; Passenberg, C.F.; Tanzmeister, G.; Wollherr, D.; Buss, M. Interactive scene prediction for auto-motive applications. In Proceedings of the 2013 IEEE Intelligent Vehicles Symposium (IV), Gold Coast, Australia, 23–26 June 2013; pp. 1028–1033.
37. Schlechtriemen, J.; Wirthmueller, F.; Wedel, A.; Breuel, G.; Kuhnert, K.-D. When will it change the lane? A probabilistic regression approach for rarely occurring events. In Proceedings of the 2015 IEEE Intelligent Vehicles Symposium (IV), Seoul, Korea, 28 June–1 July 2015; pp. 1373–1379.
38. Gonzalez, D.S.; Dibangoye, J.S.; Laugier, C. High-speed highway scene prediction based on driver models learned from demonstrations. In Proceedings of the 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, Brazil, 1–4 November 2016; pp. 149–155.
39. Mozaffari, S.; Al-Jarrah, O.Y.; Dianati, M.; Jennings, P.; Mouzakitis, A. Deep Learning-Based Vehicle Behavior Prediction for Autonomous Driving Applications: A Review. *IEEE Trans. Intell. Transp. Syst.* **2020**, 1–15. [[CrossRef](#)]
40. Lee, D.; Kwon, Y.P.; McMains, S.; Hedrick, J.K. Convolution neural network-based lane change intention prediction of surrounding vehicles for ACC. In Proceedings of the 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama, Japan, 16–19 October 2017; pp. 1–6.
41. Cui, H.; Radosavljevic, V.; Chou, F.-C.; Lin, T.-H.; Nguyen, T.; Huang, T.-K.; Schneider, J.; Djuric, N. Multimodal Trajectory Predictions for Autonomous Driving using Deep Convolutional Networks. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 2090–2096.
42. Hoermann, S.; Bach, M.; Dietmayer, K. Dynamic Occupancy Grid Prediction for Urban Autonomous Driving: A Deep Learning Approach with Fully Automatic Labeling. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 2056–2063.
43. Luo, W.; Yang, B.; Urtasun, R. Fast and Furious: Real Time End-to-End 3D Detection, Tracking and Motion Forecasting with a Single Convolutional Net. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 3569–3577.
44. Casas, S.; Luo, W.; Urtasun, R. Intentnet: Learning to predict intention from raw sensor data. In Proceedings of the 2nd Conference on Robot Learning, Zürich, Switzerland, 29–31 October 2018; pp. 947–956.
45. Zhao, T.; Xu, Y.; Monfort, M.; Choi, W.; Baker, C.; Zhao, Y.; Wang, Y.; Wu, Y.N. Multi-Agent Tensor Fusion for Contextual Trajectory Prediction. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 12118–12126.
46. Lee, N.; Choi, W.; Vernaza, P.; Choy, C.B.; Torr, P.H.S.; Chandraker, M. DESIRE: Distant Future Prediction in Dynamic Scenes with Interacting Agents. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 22–25 July 2017; pp. 2165–2174.
47. Schreiber, M.; Hoermann, S.; Dietmayer, K. Long-Term Occupancy Grid Prediction Using Recurrent Neural Networks. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 9299–9305.
48. Deo, N.; Trivedi, M.M. Multi-Modal Trajectory Prediction of Surrounding Vehicles with Maneuver based LSTMs. In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 26–30 June 2018; pp. 1179–1184.
49. Tian, Y.; Pan, L. Predicting Short-Term Traffic Flow by Long Short-Term Memory Recurrent Neural Network. In Proceedings of the 2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity), Chengdu, China, 19–21 December 2015; pp. 153–158. [[CrossRef](#)]
50. Zambrano-Martinez, J.L.; Calafate, C.T.; Soler, D.; Lemus-Zúñiga, L.-G.; Cano, J.-C.; Manzoni, P.; Gayraud, T. A central-ized route-management solution for autonomous vehicles in urban areas. *Electronics* **2019**, *8*, 722. [[CrossRef](#)]
51. Bahdanau, D.; Cho, K.; Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv* **2014**, arXiv:1409.0473.

52. Luong, M.; Pham, H.; Manning, C.D. Effective Approaches to Attention-Based Neural Machine Translation. *arXiv* **2015**, arXiv:1508.04025.
53. Lin, L.; Li, W.; Bi, H.; Qin, L. Vehicle Trajectory Prediction Using LSTMs with Spatial-Temporal Attention Mechanisms. *IEEE Intell. Transp. Syst. Mag.* **2021**. [[CrossRef](#)]
54. Messaoud, K.; Yahiaoui, I.; Verroust-Blondet, A.; Nashashibi, F. Attention Based Vehicle Trajectory Prediction. *IEEE Trans. Intell. Veh.* **2021**, *6*, 175–185. [[CrossRef](#)]
55. Kim, H.; Kim, D.; Kim, G.; Cho, J.; Huh, K. Multi-Head Attention based Probabilistic Vehicle Trajectory Prediction. In Proceedings of the 2020 IEEE Intelligent Vehicles Symposium (IV), Las Vegas, NV, USA, 19 October–13 November 2020; pp. 1720–1725.
56. James Colyar, J.H. *US Highway 101 Dataset*; FHWA-HRT07-030; Federal Highway Administration (FHWA): Washington, DC, USA, 2007.
57. James Colyar, J.H. *US Highway i-80 Dataset*; FHWA-HRT-06-137; Federal Highway Administration (FHWA): Washington, DC, USA, 2006.
58. Yoon, Y.; Kim, T.; Lee, H.; Park, J. Road-Aware Trajectory Prediction for Autonomous Driving on Highways. *Sensors* **2020**, *20*, 4703. [[CrossRef](#)]
59. Tijerina, L.; Garrott, W.; Glecker, M.; Stoltzfus, D.; Parmer, E. *Van and Passenger Car Driver Eye Glance Behavior during Lane Change Decision Phase, Interim Report*; Transportation Research Center Report; National Highway Transportation Safety Administration: Washington, DC, USA, 1997.
60. Toledo, T.; Zohar, D. Modeling Duration of Lane Changes. *Transp. Res. Rec. J. Transp. Res. Board* **2007**, *1999*, 71–78. [[CrossRef](#)]
61. Lee, S.E.; Olsen, E.C.; Wierwille, W.W. *A Comprehensive Examination of Naturalistic Lane-Changes; United States*; National Highway Traffic Safety Administration: Washington, DC, USA, 2004.
62. Hanowski, R.J. *The Impact of Local/Short Haul Operations on Driver Fatigue*; Virginia Polytechnic Institute and State University: Montgomery County, MD, USA, 2000.
63. Neubig, G. Neural machine translation and sequence-to-sequence models: A tutorial. *arXiv* **2017**, arXiv:1703.01619 2017.
64. Sun, C.; Shrivastava, A.; Singh, S.; Gupta, A. Revisiting unreasonable effectiveness of data in deep learning era. In Proceedings of the 16th IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 843–852.
65. Sarker, I.H. Machine Learning: Algorithms, Real-World Applications and Research Directions. *SN Comput. Sci.* **2021**, *2*, 160. [[CrossRef](#)] [[PubMed](#)]
66. Phillips, D.J.; Wheeler, T.A.; Kochenderfer, M.J. Generalizable intention prediction of human drivers at intersections. In Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV), Los Angeles, CA, USA, 11–14 June 2017; pp. 1665–1670.
67. Palazzi, A.; Solera, F.; Calderara, S.; Alletto, S.; Cucchiara, R. Learning where to attend like a human driver. In Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV), Los Angeles, CA, USA, 11–14 June 2017; pp. 920–925.
68. Ponziani, R.L. *Turn Signal Usage Rate Results: A Comprehensive Field Study of 12,000 Observed Turning Vehicles*; SAE Technical Paper; SAE International: Warrendale, PA, USA, 2012. [[CrossRef](#)]
69. Faw, H.W. To signal or not to signal: That should not be the question. *Accid. Anal. Prev.* **2013**, *59*, 374–381. [[CrossRef](#)]
70. Park, C.; Jeong, N.-T.; Yu, D.; Hwang, S.-H. Path Generation Algorithm Based on Crash Point Prediction for Lane Changing of Autonomous Vehicles. *Int. J. Automot. Technol.* **2019**, *20*, 507–519. [[CrossRef](#)]